

Review

Localization and Mapping for Self-Driving Vehicles: A Survey

Anas Charroud ^{1,*}, Karim El Moutaouakil ^{2,†}, Vasile Palade ^{3,†}, Ali Yahyaouy ^{4,†}, Uche Onyekpe ^{3,5,†} and Eyo U. Eyo ^{6,†}

¹ Technical Sciences Faculty, Sidi Mohamed Ben Abdellah University, Fès-Atlas 30000, Morocco

² Laboratory of Engineering Sciences, Multidisciplinary Faculty of Taza, Sidi Mohamed Ben Abdellah University, Taza 35000, Morocco; karim.elmoutaouakil@usmba.ac.ma

³ Centre for Computational Science and Mathematical Modelling, Coventry University, Priory Road, Coventry CV1 5FB, UK; vasile.palade@coventry.ac.uk (V.P.); uche.onyekpe@ofcom.org.uk (U.O.)

⁴ Computer Science, Signals, Automatics and Cognitivism Laboratory, Sciences Faculty of Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fès-Atlas 30000, Morocco; ali.yahyaouy@usmba.ac.ma

⁵ Office of Communications, 15 Lauriston Place, Edinburgh EH3 9EP, UK

⁶ School of Engineering, College of Arts, Technology and Environment, University of the West of England, Bristol BS16 1QY, UK; eyo.eyo@uwe.ac.uk

* Correspondence: anas.charroud@usmba.ac.ma

† These authors contributed equally to this work.

Abstract: The upsurge of autonomous vehicles in the automobile industry will lead to better driving experiences while also enabling the users to solve challenging navigation problems. Reaching such capabilities will require significant technological attention and the flawless execution of various complex tasks, one of which is ensuring robust localization and mapping. Recent surveys have not provided a meaningful and comprehensive description of the current approaches in this field. Accordingly, this review is intended to provide adequate coverage of the problems affecting autonomous vehicles in this area, by examining the most recent methods for mapping and localization as well as related feature extraction and data security problems. First, a discussion of the contemporary methods of extracting relevant features from equipped sensors and their categorization as semantic, non-semantic, and deep learning methods is presented. We conclude that representativeness, low cost, and accessibility are crucial constraints in the choice of the methods to be adopted for localization and mapping tasks. Second, the survey focuses on methods to build a vehicle's environment map, considering both the commercial and the academic solutions available. The analysis proposes a difference between two types of environment, known and unknown, and develops solutions in each case. Third, the survey explores different approaches to vehicle localization and also classifies them according to their mathematical characteristics and priorities. Each section concludes by presenting the related challenges and some future directions. The article also highlights the security problems likely to be encountered in self-driving vehicles, with an assessment of possible defense mechanisms that could prevent security attacks in vehicles. Finally, the article ends with a debate on the potential impacts of autonomous driving, spanning energy consumption and emission reduction, sound and light pollution, integration into smart cities, infrastructure optimization, and software refinement. This thorough investigation aims to foster a comprehensive understanding of the diverse implications of autonomous driving across various domains.

Keywords: autonomous driving; feature extraction; mapping; localization; automotive security; SLAM



Citation: Charroud, A.; El Moutaouakil, K.; Palade, V.; Yahyaouy, A.; Onyekpe, U.; Eyo, E.U. Localization and Mapping for Self-Driving Vehicles: A Survey. *Machines* **2024**, *12*, 118. <https://doi.org/10.3390/machines12020118>

Academic Editors: Akos Odry, Peter Sarcevic and Jozsef Sarosi

Received: 11 December 2023

Revised: 25 January 2024

Accepted: 26 January 2024

Published: 7 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the last decade, self-driving vehicles have shown impressive progress, with many researchers working in different laboratories and companies and experimenting in various environmental scenarios. Some of the questions in the minds of potential users relate to the advantages that are offered by this technology and whether we can rely on it or

not. To respond to these questions, one needs to look at the numbers of accidents and deaths registered daily from non-autonomous vehicles. According to the World Health Organization's official website, approximately 1.3 million people, mainly children and young adults, die yearly in vehicle crashes. Most of these accidents (93%) are known to occur in low- and middle-income countries [1]. Human behaviors, including traffic offenses, driving while under the influence of alcohol or psychoactive substances, and late reactions, are the major causes of crashes. Consequently, self-driving technology has been conceived and developed to replace human-vehicle interaction. Despite some of the challenges of self-driving vehicles, the surveys in [2] indicate that about 84.4% of people are in favor of using self-driving vehicles in the future. Sale statistics also show that the worth of these vehicles was USD 54 billion in 2021 [3]. This outcome demonstrates the growing confidence in the usage of this technology. It has also been suggested that this technology is capable of reducing pollution, energy, time consumption, and accident rates, while also ensuring the safety and comfort of its users [4].

The Society of Automotive Engineers (SAE) divides the technical revolution resulting from this technology into five levels. The first two levels do not provide many automatic services and, therefore, have a low risk potential. The challenge starts from level three, where the system is prepared to avoid collisions and has assisted steering, braking, etc. Moreover, humans must be prudent with respect to any hard decision or alert from the system. Levels four and five refer to vehicles that are required to take responsibility for driving without any human interaction. Hence, these levels are more complex, especially if one considers how dynamic and unpredictable the environment can be, where there may be a chance of collision from anywhere. Ref. [4] affirms that no vehicle has achieved robust results in these levels at the time of writing this paper.

Creating a self-driving vehicle is a complicated task, where it is necessary to perform predefined steps consistently and achieve the stated goals in a robust manner. The principal question that should be answered is: how can vehicles work effectively without human interaction? Vehicles need to continuously sense the environment; hence, they should be equipped with accurate sensors, like Global Navigation Satellite System (GNSS), Inertial Measurement Unit (IMU), wheel odometry, LiDAR, radar, cameras, etc. Moreover, a high-performance computer is needed to handle the immense volume of data collected and accurately extract relevant information. This information helps the vehicles make better decisions and execute them using specific units prepared for this purpose.

To achieve autonomous driving, Ref. [5] proposed an architecture that combines localization, perception, planning, control, and system management. Localization is deemed an indispensable step in the operation autonomous systems since we need accurate information of its location. Figure 1 illustrates the importance of localization and mapping steps in the deployment of other self-driving tasks, like path planning (e.g., finding the shortest path to a certain destination), ensuring reliable communications with other AVs, and performing necessary tasks at the right moment, such as overtaking vehicles, breaking, or accelerating. A small localization error can lead to significant issues, like traffic perturbation, collisions, accidents, etc.

Finding positions can be implemented with a 3D vector, with lateral, longitudinal, and orientation components called "heading":

$$x_t = [a_t, b_t, \theta_t]$$

where a_t and b_t are the Cartesian coordinates of the vehicle and θ_t is the rotation of it. There exist other representations using speed, acceleration, etc., see [6] for more details. However, we use the above representation most of the time, where the height is generally equal to 0, as there is no rotation about x -axis and y -axis. This vector is denoted by the state vector x_t .

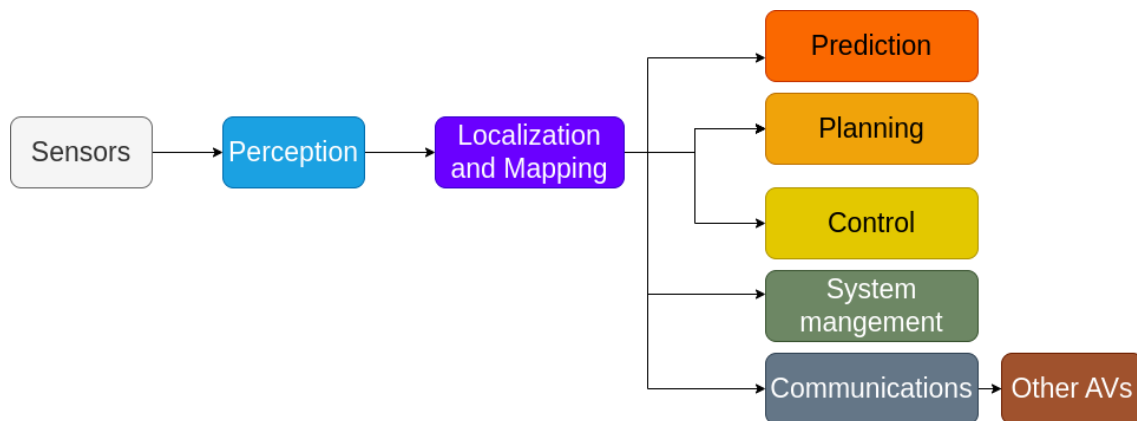


Figure 1. Steps to follow to achieve a self-driving vehicle.

One of the classical methods is based on the Global Navigation Satellite Systems (GNSS). This technique uses the trilateration approach, which allows the detection of the position anywhere in the world using three satellites. Moreover, this is a cheap solution. However, GNSS-based methods may suffer from signal blockage, and they are not preferred in some environments, especially when an obstacle cuts the line of sight. In addition, the errors of positioning exceed 3 m. All these issues can lead to unreliability and unsafety in driving. Several papers in the literature have attempted to address this problem. For example, the D-GPS or RTK-based technologies [7].

Localization based on wheels odometry is another approach that can be used to find the poses that represents the position based on a known starting point and the distance traveled. This alternative method can be applied without external references and utilized in other autonomous systems not only in self-driving vehicles. However, the method suffers from cumulative errors provided by wheel slips [7].

Inertial Navigation System (INS) uses the data provided by the IMU sensor, including accelerometers, gyroscopes, and magnetometers, and the dead-reckoning to localize the vehicle without any external references. But, that is still very restricted and limited because of cumulative errors [7].

The weakness of the traditional methods is that they work only with one source of information, and their performance depends relatively on the structure of the environment. Due to these limitations, researchers have attempted to combine information from various sensors in order to explore the advantages provided by each of them and tackle the above mentioned issues. We found that the subject of localization and mapping is widely treated and surveyed by other previous articles in the literature. Table 1 presents our effort to put together the previous surveys that investigated the same subject. We have identified some concepts that must be investigated in relation to the localization and mapping in autonomous vehicles, which include feature extraction, mapping, ego-localization (vehicle localization based on its own sensors), co-localization (vehicle localization based on its own sensors and nearby vehicles), simultaneous localization and mapping (SLAM), security of autonomous vehicles, environmental impact, and finally we present the challenges and future directions in the area.

Table 1. Comparison of previous surveys on localization and mapping.

Survey	Sensors	Features Extraction	Mapping	Ego-Localization	Co-Localization	SLAM	Security Autonomous	Environmental Impact	Challenges and Future Directions
[4]	GPS/INS Cameras LiDAR	Middle	Middle	Middle	No	Middle	Limited	No	Limited
[8]	Cameras	Extensive	Limited	Extensive	No	Middle	No	No	Middle
[9]	Cameras	Extensive	Limited	Extensive	No	Middle	No	No	Middle
[10]	Cameras LiDAR	Limited	Limited	Extensive	No	Limited	No	No	Extensive
[11]	LiDAR	Extensive	No	No	No	No	No	No	Extensive
[12]	LiDAR	Extensive	No	No	No	No	No	No	Extensive
[13]	Cameras LiDAR	Limited	Extensive	Extensive	No	Extensive	No	No	Extensive
[14]	Cameras LiDAR	Limited	Extensive	Extensive	No	Extensive	No	No	Extensive
[15]	Cameras	Limited	Extensive	Extensive	No	Extensive	No	No	Extensive
[16]	Cameras LiDAR	Limited	Extensive	Extensive	No	Extensive	No	No	Extensive
[17]	Cameras LiDAR	Limited	Extensive	Extensive	Extensive	Extensive	No	No	Extensive
[18]	All	No	Middle	Extensive	Extensive	Limited	Limited	No	Limited
[19]	Cameras LiDAR	Extensive	Extensive	Extensive	No	Limited	No	No	Limited
[20]	GPS/INS Cameras LiDAR	Limited	Limited	Extensive	No	Limited	No	No	Extensive
Ours	GPS/INS Cameras LiDAR	Extensive	Extensive	Extensive	Extensive	Extensive	Extensive	Extensive	Extensive

The surveys [8,9] studied visual-based localization methods, where they investigated a couple of methods of data representation and feature extraction. They deeply discussed the localization techniques as well as the challenges that can be met by these techniques. However, Light Detection and Ranging (LiDARs) methods are neglected in these two surveys. Grigorescu et al. [10] restricted their survey to only deep learning methods. Similarly, Refs. [11,12] have surveyed 3D object detection methods and feature extraction methods in points cloud datasets. On the other hand, many articles [13–16] have tackled the problem of Simultaneous Localization and Mapping (SLAM) and have listed recent works previously elaborated in this field. These surveys lack information on the feature extraction, and they did not present much information about the existing types of maps. An interesting survey was performed in [17], where they analyzed different methods that can be applied to Vehicular Ad-hoc Network (VANET) localization, but without tackling neither feature extraction nor mapping methods. Like the survey we present here, Kuutti et al. [18] have investigated the positioning methods according to each sensor. Moreover, they have emphasised the importance of cooperative localization, by reviewing the Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) localization. However, these surveys did not present any feature extraction methods. Badue et al. [19] surveyed extensively the methods that aim to solve the localization and mapping problem in self-driving cars. However, issues related to the protection and security of autonomous vehicles have not been investigated. The survey in [20] covered the localization and mapping problem for

racing autonomous vehicles. Based on these limitations in the previous surveys that can be found in the literature, we put forward here a survey that consists of the following main contributions:

- Comprehensively reviewing the feature extraction methods in the field of autonomous driving.
- Providing clear details on what are the solutions to create a map for autonomous vehicles, by investigating the solutions that use pre-built maps, which can be pre-existing commercial maps or generated by some academic solutions, or the online map solution, which creates a map in the same time with the positioning process.
- Providing necessary background information and demonstrating many existing solutions to solve the localization and mapping tasks.
- Reviewing and analysing localization approaches that exist in the literature together with the related solutions to ensure data security.
- Exploring the environmental footprint and impact of self-driving vehicles in relation to localization and mapping techniques and unveiling solutions for this.

The rest of this paper is organized as follows: Section 2 presents different methods to extract relevant features, precisely from LiDAR and camera sensors. Section 3 depicts relevant tips to generate an accurate map. Section 4 addresses the recent localization approaches in the literature. Section 5 highlights the levels of possible attacks and some defense mechanisms for localization security. Section 6 discusses the environmental impact of self-driving vehicle. Section 7 draws a synthetic conclusion and provides a perspective for future work in the area.

2. Feature Extraction

Feature extraction is the process of providing a relevant representation of the raw data. In other words, it is manipulating the data to be easy to learn while being more robust to generalize. It is the same here with sensors' data. Extracting suitable information is very useful, and it will reduce cost, energy and aid the models to be fast and accurate in the execution. In this section, we give an overview of what was done previously in the area. However, we limit the discussions to two sensors, namely, LiDAR and cameras, because we believe that they are more suitable and useful in practice. Figure 2 provides a clear flowchart to understand these features.

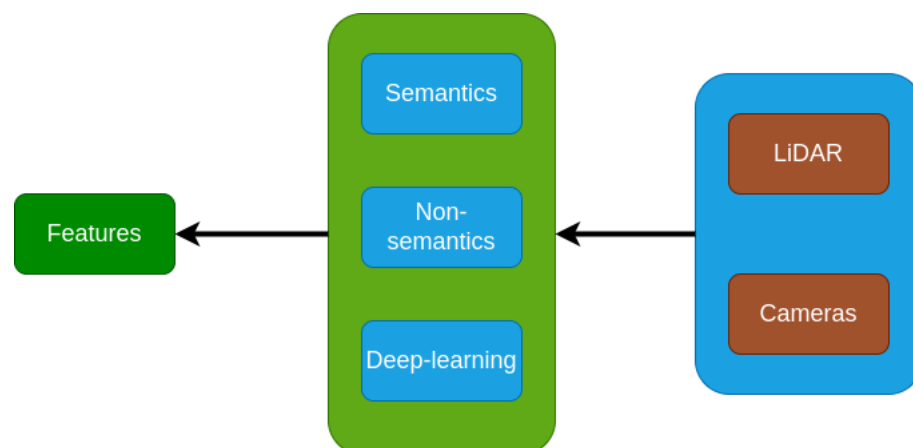


Figure 2. Flowchart of feature extraction.

2.1. Semantic Features

Semantic features essentially deals with extracting a known target (poles, facades, curbs, etc.) from the raw data, in order to localize the vehicle more accurately.

LiDAR features: We can extract lane markings using specialized methods that differentiate lanes from the surrounding environment. These techniques analyze how LiDAR beams reflect off surfaces and consider variations in elevation to accurately separate lane

boundaries. In [21], authors extract lane marking as a feature to localize the vehicle by using the Radon transform to approximate polylines of the lane. Then, they have applied the Douglas-Peukers algorithm to refine the shape points that estimate the road center lines. In the work carried out by Im et al. [22], attempts were made to detect two kinds of features. The first category was produced using LiDAR reflectivity from layer 17 to 32 to extract road marking. Indeed, a binarization layer was applied on the extracted ground plane. This binarization is done by using the Otsu thresholding method [23]. Finally, they used the Hough transform to infer road marking lines. On the other hand, the second category focuses on the extraction of the building walls because they resemble lines in the 2D horizontal plane. So, they have projected all these features into a 2D probabilistic map. However, the existence of the poles like street trees affects the creation of the map of lines. To solve this problem, unnecessary structures were eliminated by applying the Iterative-End-Point-Fit (IEPF) algorithm. This is considered as one of the most reliable algorithms for line fitting, and it gives the best accuracy in a short calculation time. Zhang et al. [24] explored the height of curbs between 10–15 cm, and the smoothness and continuity of points cloud around the curbs to propose a method to track the curbs. This approach is also used in [25] with some modifications. The existence of curbs on almost every road could be regarded as very useful. The authors of [26] used the combination of information from LiDAR and cameras to extract poles like traffic signs, trees, and street lamps, since they are more representative for localization purposes. The authors projected, in each layer, the point clouds in a 2D grid map after removing the ground plane, and they assumed that the connected grid cell of points, at the same height, is supposed to be a candidate object. To ensure the reliability of this method, they have created a cylinder for each landmark candidate and form the poles' shapes. From the camera, they attempted to calculate a dense disparity mapping and apply pattern recognition on the result of the map to detect landmarks. Kummerle et al. [27] adopts the same idea as [26], where they attempted to extract poles and facades from the LiDAR scans. They also detected road marking by a stereo camera. In [28], the authors extracted, as the first kind of such features, building facades, traffic poles, and trees, by calculating the variance of each voxel, then, they checked the size of a fixed threshold, and concluded on the closeness of the points. Those voxels are grouped and arranged in the same vertical direction. Hence, each cluster that respects the variance condition in each voxel is considered as a landmark. The second kind is interested in the reflectivity of the ground, which is extracted with the RANSAC algorithm from LiDAR intensity. A performant algorithm to detect poles was proposed in [29]. Firstly, a 3D voxel map was applied on the set of points cloud; then, they eliminated cells with fewer points. Thereafter, they determined the cluster's boundaries by calculating the intensity of points in the pole's shape candidate to deduce the highest and lowest parts. Finally, they checked if the points clouds in the entire core of the pole candidates did satisfy the density condition to be considered as pole candidates. Ref. [30] implemented a method with three steps: voxelization, horizontal clustering, and vertical clustering (voxelization is a preprocessing step). Typically, it is intended to remove the ground plan and regroup each part of the points cloud into voxels; then, pick the voxels with the number of points greater than a fixed threshold. Also, they exploited the pole characteristics in an urban area, like the isolation of the poles, distinguishable from the surrounding area, in order to extract horizontal and vertical features. Ref. [31] is an interesting article that provides highly accurate results of detecting pole landmarks. Indeed, the authors have used a 3D probabilistic occupancy grid map based mainly on the Beta distribution applied for each layer. Then, they have calculated the difference between the average occupancy value inside the pole and the maximum occupancy value of the surrounding poles, which is a value between $[-1, 1]$. Greater value means higher probability that it is a pole. A GitHub link for the method was provided in [31]. Ref. [32] is another method that uses the pole landmarks, providing a high-precision method to localize self-driving vehicles. The method assumed five robust conditions: common urban objects, time-invariant location, time-invariant appearance, viewpoint-invariant observability, and high frequency of occurrence. First, they

removed the ground plan using the M-estimator Sample Consensus (MSAC). Secondly, they performed an occupied grid map, which isolates the empty spaces. Afterwards, they performed clustering with the connected component analysis (CCA) to group the connected occupied grid into an object. Finally, PCA was used to infer the position of the poles. The idea is to use the three eigenvalues $\lambda_x, \lambda_y, \lambda_z$ obtained in these three conditions (s_{th} is a fixed threshold).

$$\begin{aligned} \lambda_z &\gg \lambda_x, \lambda_y > 0 \\ \lambda_x &\approx \lambda_y \\ \frac{\lambda_z}{\min(\lambda_x, \lambda_y)} &\leq s_{th} \end{aligned}$$

These conditions formed precisely the dimensions of the poles. The article [33] has detected three kinds of features. First, planar features, like walls, building facades, fences, and other planar surfaces. Second, pole features include streetlights, traffic signs, tree trunks. Third, curbs shapes. After removing the ground plan using the Cloth Simulation Filter, the authors have concluded that all landmarks are vertical and higher above the ground plan and contain more points than its surrounding. They filtered cells that do not respect the above conditions. Planar features and curb features are considered as lines that could be approximated by a RANSAC algorithm or running a Hough transform. This research focuses on the use of pole-like structures as essential landmarks for accurate localization of autonomous vehicles in urban and suburban areas. A novel approach is to create a detailed map of the poles, taking into account both geometric and semantic information. Ref. [34] improves the localization of autonomous vehicles in urban areas using bollard-like landmarks. A novel approach integrates geometric and semantic data to create detailed maps of bollards using a mask-rank transformation network. The proposed semantic particle filtering improves localization accuracy, validated on the Semantic KITTI dataset. Integrating semantics into maps is essential for robust localization of autonomous vehicles in uncertain urban environments. Ref. [35] introduces a novel method for accurate localization of mobile autonomous systems in urban environments, utilizing pole-like objects as landmarks. Range images from 3D LiDAR scans are used for efficient pole extraction, enhancing reliability. The approach stands out for its improved pole extraction, efficient processing, and open-source implementation with a dataset, surpassing existing methods in various environments without requiring GPU support.

Camera features. Due to its low cost, and lightweight, the camera is an important sensor widely used by researchers and automotive companies. Ref. [13] divided cameras into 4 principal types: monocular camera, stereo camera, RGB-D camera, and event camera. Extracting features from the camera sensor is another approach that should be investigated. Ref. [8] is one of the best works done in visual-based localization, where the authors have given an overall idea about the recent trend in this field; they divide features into local and global features. Local features search to extract precise information from a local region in the image, which is more robust for the generalization task and more stable with image variations. Secondly, global features focus on the extraction of features from the whole image. These features could either be semantical or not. Most researchers working with semantics try to extract contours, line segments, and objects in general. In [36], the authors extracted two types of linear features, edges, and ridges by using a Hough transform. Similarly, authors of [37] have used a monocular camera to extract lanes' markings and pedestrian crossing lines. Polylines approximate these features. On the other hand, Ref. [38] performed a visual localization using cameras, where they brought out edges from environment scenes as a features/input. Each image input is processed as follows: the first step consists of solving the radial distortion issue, which is the fact that straight lines bend into circular arcs or curves [39]. This problem can be solved by the radial distortion correction methods and projective transformation into the bird's eye view. Moreover, they calculated the gradient magnitude, which creates another image, called gradient image, split into several intersecting regions. After that, a Hough transform is applied to refine the edges of the line segments. Finally, a set of conditions depending on

geometric and connectivity of the segment was checked to find the last representation of the edge polylines. Localization of mobile devices was investigated in [40] by using features that satisfy some characteristics, such as: permanent (statical), informative (distinguishable from others), widely available. From the features we can derive: alignment tree, autolib station, water fountain, street lights, traffic lights, etc. Reference [41] introduced MonoLiG, a pioneering framework for monocular 3D object detection that integrates LiDAR-guided semi-supervised active learning (SSAL). The approach optimizes model development using all data modalities, employing LiDAR to guide the training without inference overhead. The authors utilize a LiDAR teacher, monocular student cross-modal framework for distilling information from unlabeled data, augmented by a noise-based weighting mechanism to handle sensor differences. They propose a sensor consistency-based selection score to choose samples for labeling, which consistently outperforms active learning baselines by up to 17% in labeling cost savings. Extensive experiments on KITTI and Waymo datasets confirm the framework's effectiveness, achieving top rankings in official benchmarks and outperforming existing active learning approaches.

2.2. Non-Semantics Features

Non-semantics features are unlike the semantics once; they do not have any significance in their contents. They provide an abstract scan without adopting any significant structure like poles, buildings, etc. This gives a more general representation of the environment and reduces execution time instead of searching about a specific element that can not exist everywhere.

LiDAR features. In [42], the method consists of four main steps. They started with pre-processing, which aligns each local neighborhood to a local reference frame using Principal Component Analysis (PCA). The smallest principal component is taken as a normal (perpendicular to the surface). Secondly, Pattern Generation is where the remaining local neighborhood points are transformed from 3D into 2D within a grid map; each cell contains the maximum value of reflectivity. Furthermore, the descriptor calculation is performed by the DAISY descriptor, which works by convolving the intensity pattern by a Gaussian kernel. After that, a gradient of intensities is calculated for eight radial orientations. Moreso, a smoothening is performed using Gaussian kernel of three different standard deviations. Finally, a normalization step is applied to maintain the value of the gradients within the descriptor. 'GRAIL' [42] is able to compare the query input with twelve distinctive shapes that can be used as relevant features for localization purposes. Hungar et al. [43] used a non-semantical approach where a reduction of the time of execution, by selecting points whose sphere of the local neighborhood with radius r , including an amount of point that will exceed a fixed threshold, was carried out. After that, they distinguish the remaining patterns into curved and non-curved by using k -medoids clustering. The authors use a DBSCAN clustering to aggregate similar groups to infer features. Lastly, the creation of the key features and map features relied on different criteria, including distinctiveness, uniqueness, spatial diversity, tracking stability, persistence. A 6D pose estimation was performed by [44], where they describe the vehicle's roadside and considered it a useful feature for the estimation model. To achieve this, they proposed preprocessing the point cloud through ROI (Region of Interest) filtering in order to remove long-distance background points. After that, a RANSAC algorithm was performed to find the corresponding equation of the road points. Thanks to the Radius Outlier Removal filter, which the authors used to reduce the noise by removing isolated points and reduce their interference. Meanwhile, the shape of the vehicles was approximated with the help of the Euclidean clustering algorithm presented in [45]. Charroud et al. [46,47] have removed the ground plan of all the LiDAR scan to reduce a huge amount of points, and they have used a Fuzzy K-means clustering technique to extract relevant features from the LiDAR scan. An extension of this work [48] adds a downsampling method to speed up the calculation process of the Fuzzy K-means algorithm.

Descriptors based methods can be considered an interesting idea since they are widely used to extract meaningful knowledge directly from a set of points' cloud. These methods do enable a separation of each feature point regarding the perturbation caused by noise and varying density, and the change of the appearance of the local region [49]. They also added that four main criteria should be used while performing features description, citing the descriptiveness, robustness, efficiency, and compactness. These criteria involved the reliability, the representativity, the cost of time and the storage space. These descriptors can be used in the points pair-based approach, which is the task of exploiting any relationship between two points like distance or angle between normals (the perpendicular lines to a given object), or boundary to boundary relations, or relations between two lines [50]. Briefly, with respect to this methods, we can cite CoSPAIR [51], PPFH [52], HoPPF [53], PPTFH [54]. Alternatively, they can be included to extract local features. For example, TOLDI [55], BRoPH [49], and 3DHoPD [56].

Camera features. Extracting non-semantic features from the camera sensor is widely treated in the literature. However, this article focuses on methods that work to find pose estimation. Due to the fastness of the ORB compared to other descriptors like SIFT or SURF, many researchers have adopted this descriptor to represent image features. For this purpose, Ref. [57] attempted to extract key features by using an Oriented FAST and rotated BRIEF (ORB) descriptor, to achieve an accurate matching between the extracted map and the original map. Another matching features' method was proposed in [58], where the authors sort to extract holistic features from front view images by using the ORB layer and BRIEF layer descriptor to find a great candidate node, while local features from downward view images were detected using FAST-9, which are fast enough to cope with this operation. Gengyu et al. [59] extracted ORB features and converted them to visual words based on the DBow3 approach [60,61] used ImageNet-trained Convolutional Neural Networks (CNN) features (more details in the next sub-section) to extract object landmarks from images, as they are considered to be more powerful for localization tasks. Then, they implement Scale-Invariant Feature Transform (SIFT) for further improvement of the features.

2.3. Deep Learning Features

It is worth mentioning the advantage of working with deep learning methods, which try to imitate the working process of the human brain. Deep learning (DL) is widely encountered in various application domains, such as medical, financial, molecular biology, etc. It can be used for different tasks, such as object detection and recognition, segmentation, classification, etc. In the followings, we survey recent papers that use DL for vehicle localization and mapping.

LiDAR and/or Camera features. One of the most interesting articles done in this field is [11], where the authors provided an overall picture of object detection with LiDAR sensors using deep learning. This survey divides the state-of-art algorithms into three parts: projection-based methods, which project the set of points cloud into a 2D map respecting a specific viewpoint; voxel-based methods, which involve the task of making data more structured and easier to use by discretizing the space into a fixed voxel grid map; finally, point-based methods, which work directly on the set of points' cloud. Our literature investigation concludes that most of the methods are based on the CNN architecture with different modifications in the preprocessing stage.

- **MixedCNN-based Methods:** Convolutional Neural Network (CNN) is one of the most common methods used in computer vision. These types of methods use mathematical operations called 'convolution' to extract relevant features [62]. VeloFCN [63] is a projection-based method and one of the earliest methods that uses CNN for 3D vehicle detection. The authors used a three convolution layer structure to down-sample the input front of the view map, then up-sample with a deconvolution layer. The output of the last procedure is fed into a regression part to create a 3D box for each pixel. Meanwhile, the same results were entered for classification to check if the corresponding pixel was a vehicle or not. Finally, they grouped all candidates'

boxes and filtered them by a Non-Maximum Suppression (NMS) approach. In the same vein, LMNet [64] increased the zone of detection to find road objects by taking into consideration five types of features: reflectance, range, distance, side, height. Moreover, they change the classical convolution by the dilated ones. The Voxelnet [65] method begins with the process of voxelizing the set of points cloud and passing it through the VFE network (explained below) to obtain robust features. After that, a 3D convolutional neural network is applied to group voxels features into a 2D map. Finally, a probability score is calculated using an RPN (Region Proposal Network). The VFE network aims to learn features of points by using a multi-layer-perceptron and a max-pooling architecture to obtain point-wise features. This architecture concatenates features from the MLP output and the MLP + Maxpooling. This process is repeated several times to facilitate the learning. The last iteration is fed to an FCN to extract the final features. BirdNet [66] generates a three-channel bird eye's view image, which encodes the height, intensity, and density information. After that, a normalization was performed to deal with the inconsistency of the laser beams of the LiDAR devices. BirdNet uses a VGG16 architecture to extract features, and they adopt a Fast-RCNN to perform object detection and orientation. BirdNet+ [67] is an extension of the last work, where they attempted to predict the height and vertical position of the centroid object in addition to the processing of the source (BirdNet) method. This field is also approached by transfer learning, like in Complex-YOLO [68], and YOLO3D [69]. Other CNN-based method include regularized graph CNN (RGCNN) [70], Pointwise-CNN [71], PointCNN [72], Geo-CNN [73], Dynamic Graph-CNN [74] and SpiderCNN [75].

- **Other Methods:** These techniques are based on different approaches. Ref. [76] is a machine learning-based method where the authors try to voxelize the set of points cloud into 3D grid cells. They extract features just from the non-empty cells. These features are a vector of six components: mean and variance of the reflectance, three shape factors, and a binary occupancy. The authors proposed an algorithm to compute the classification score, which takes in the input of a trained SVM classification weight and features, then a voting procedure is used to find the scores. Finally, a non-maximum suppression (NMS) is used to remove duplicate detection. Interesting work is done in [77], who tried to present a new architecture of learning that directly extracts local and global features from the set of points cloud. The 3D object detection process is independent of the form of the points cloud. PointNet shows a powerful result in different situations. PointNet++ [78] extended the last work of PointNet, thanks to the Furthest Point Sampling (FPS) method. The authors created a local region by clustering the neighbor point and then applied the PointNet method in each cluster region to extract local features. Ref. [79] introduces a novel approach using LiDAR range images for efficient pole extraction, combining geometric features and deep learning. This method enhances vehicle localization accuracy in urban environments, outperforming existing approaches and reducing processing time. Publicly released datasets support further research and evaluation. The research presents PointCLIP [80], an approach that aligns CLIP-encoded point clouds with 3D text to improve 3D recognition. By projecting point clouds onto multi-view depth maps, knowledge from the 2D domain is transferred to the 3D domain. An inter-view adapter improves feature extraction, resulting in better performance in a few shots after fine-tuning. By combining PointCLIP with supervised 3D networks, it outperforms existing models on datasets such as ModelNet10, ModelNet40 and ScanObjectNNN, demonstrating the potential for efficient 3D point cloud understanding using CLIP. PointCLIP V2 [81] enhances CLIP for 3D point clouds, using realistic shape projection and GPT-3 for prompts. It outperforms PointCLIP [80] by +42.90%, +40.44%, and +28.75% accuracy in zero-shot 3D classification. It extends to few-shot tasks and object detection with strong generalization. Code and prompt details are provided. The paper [82] presents a "System for Generating 3D Point Clouds from Complex Prompts and proposes an accelerated

approach to 3D object generation using text-conditional models. While recent methods demand extensive computational resources for generating 3D samples, this approach significantly reduces the time to 1–2 min per sample on a single GPU. By leveraging a two-step diffusion model, it generates synthetic views and then transforms them into 3D point clouds. Although the method sacrifices some sample quality, it offers a practical tradeoff for scenarios, prioritizing speed over sample fidelity. The authors provide their pre-trained models and code for evaluation, enhancing the accessibility of this technique in text-conditional 3D object generation. Researchers have developed 3DALL-E [83], an add-on that integrates DALL-E, GPT-3 and CLIP into the CAD software, enabling users to generate image-text references relevant to their design tasks. In a study with 13 designers, the researchers found that 3DALL-E has potential applications for reference images, renderings, materials and design considerations. The study revealed query patterns and identified cases where text-to-image AI aids design. Bibliographies were also proposed to distinguish human from AI contributions, address ownership and intellectual property issues, and improve design history. These advances in textual referencing can reshape creative workflows and offer users faster ways to explore design ideas through language modeling. The results of the study show that there is great enthusiasm for text-to-image tools in 3D workflows and provide guidelines for the seamless integration of AI-assisted design and existing generative design approaches. The paper [84] introduces SDS Complete, an approach for completing incomplete point-cloud data using text-guided image generation. Developed by Yoni Kasten, Ohad Rahamim, and Gal Chechik, this method leverages text semantics to reconstruct surfaces of objects from incomplete point clouds. SDS Complete outperforms existing approaches on objects not well-represented in training datasets, demonstrating its efficacy in handling incomplete real-world data. Paper [85] presents CLIP2Scene, a framework that transfers knowledge from pre-trained 2D image-text models to a 3D point cloud network. Using a semantics-based multimodal contrastive learning framework, the authors achieve annotation-free 3D semantic segmentation with significant mIoU scores on multiple datasets, even with limited labeled data. The work highlights the benefits of CLIP knowledge for understanding 3D scenes and introduces solutions to the challenges of unsupervised distillation of cross-modal knowledge.

Figure 3 represents a timeline of the most popular 3D object detection algorithms.

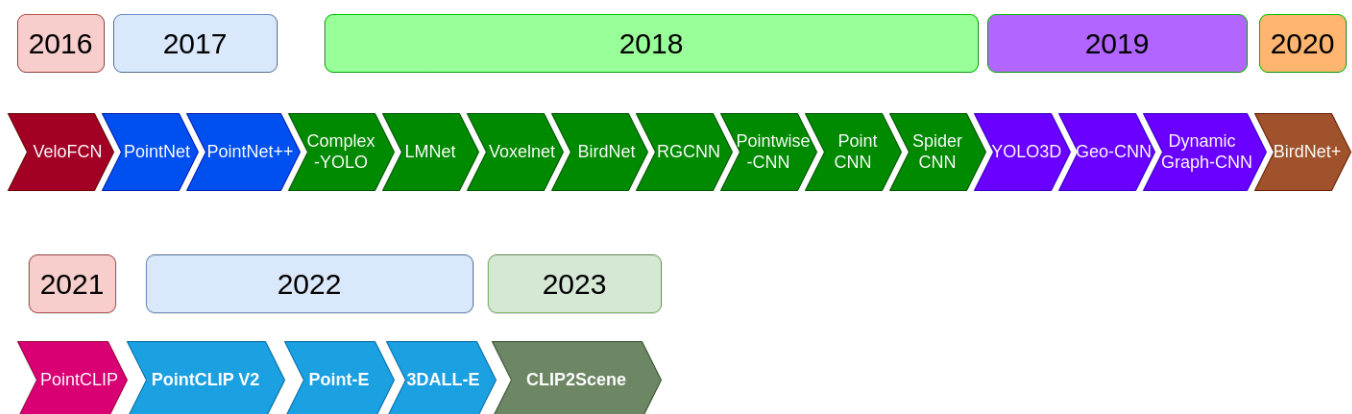


Figure 3. Timeline of 3D object detection algorithms [11].

2.4. Discussion

Table 2 provides a categorization for the surveyed papers in this section. The papers were grouped regarding the extracted features. We noticed that the papers extracted three kinds of object features from the environment: vertical, horizontal, or road curve features in case of semantic type. Also, papers have used non-semantic and deep learning methods to represent any kind of objects that exist in the environment, or to represent only a part

of them. Moreover, the table provides some methods and concepts used to extract the features. We have analysed the robustness of the extracted features to help the localization and mapping tasks by using three criteria deduced from our state-of-art investigation.

- **Time and energy cost:** being easy to detect and easy to use in terms of compilation and execution.
- **Representativeness:** detecting features that frequently exist in the environment to ensure the matching process.
- **Accessibility:** being easy to distinguish from the environment.

We have used the column 'Robustness' as a score given to each cluster of papers. The score is calculated based on the three criteria above and the analysis of experiments in the papers. According to the same table, extracting non-semantic features have the highest robustness score regarding their ability to represent the environment even with less texture, i.e., in the case of few objects in the environment like in the desert. This competence is due to the way the features are extracted. Those methods do not limit themselves to extracting one type of object. However, the map created by those features will not have a meaning. They are just helpful reference points for the localization process only.

On another hand, using semantic features helps to get a passable score to be used in localization tasks since they consume a bit more time and energy for executions because most of the time they are not isolated in the environment. One more thing is that they can not be found in any environment, which hardly affects the localization process. Despite all these negative points, these techniques reduce effectively the huge amount of points data (LiDAR or Camera) compared with the non-semantic ones. Also, those features can be used for other perception tasks.

Deep learning methods also get a passable score regarding their efficiency to represent the environment. Like the non-semantic techniques, the DL approaches ensure the representativeness of the features in all environments. However, the methods consume a lot of time and computational resources to be executed.

2.5. Challenges and Future Directions

In order to localize itself within the environment, the vehicle needs to explore the received information from the sensors. However, the huge amount of data received makes it impossible to be used in real-time localization on the vehicle, since the vehicle needs an instant interaction with the environment, e.g., accelerating, breaking, steering the wheel, etc. That is why the on-board systems need effective feature extraction methods that will distinguish relevant features for better execution of the localization process.

After surveying and analyzing related papers, the following considerations for effective localization were identified:

- Features should be robust against any external effect like weather changes, other moving objects, e.g., trees that move in the wind.
- Provide the possibility of re-use in other tasks.
- The detection system should be capable to extract features even with a few objects in the environment.
- The proposed algorithms to extract features should not hurt the system by requiring long execution time.
- One issue that should be taken into consideration is about safe and dangerous features. Each feature must provide a degree of safety (expressed as percentage), which helps to determine the nature of the feature and where they belong to, e.g., belonging to the road is safer than being on the walls.

Table 2. Categorization of the state-of-art methods that extract relevant features for localization and mapping purposes.

Paper	Features Type	Concept	Methods	Features-Extracted	Time and Energy Cost	Representativeness	Accessibility	Robustness
[21,22], [33,36], [37,38].	- Semantic	- General	- Radon transform - Douglas & Peukers algorithm - Binarization - Hough transform - Iterative-End-Point-Fit (IEPF) - RANSAC	- Road lanes - lines - ridges Edges - pedestrian crossings lines	- Consume a lot	- High	- Hard	- Passable
[24,25], [33].	- Semantic	- General	- The height of curbs between 10 cm and 15 cm - RANSAC	- Curves	- Consume a lot	- High	- Hard	- Passable
[26,27], [28,29], [30,31], [32,33], [40].	- Semantic	- Probabilistic - General	- Probabilistic Calculation - Voxelisation	- Building facades - Poles	- Consume a lot	- Middle	- Hard	- Low
[42,43], [44,46], [47,48], [51,52], [53,54], [49,55], [56,57], [58,59], [60].	- Non-Semantic	- General	- PCA - DAISY - Gaussian kernel - K-medoids - K-means - DBSCAN - RANSAC - Radius Outlier Removal filter - ORB - BRIEFAST-9	- All the environment	- Consume less	- High	- Easy	- High

Table 2. Cont.

Paper	Features Type	Concept	Methods	Features-Extracted	Time and Energy Cost	Representativeness	Accessibility	Robustness
[61,62], [63,64], [65,66], [67,68], [69,70], [71,72], [73,74], [75,76], [77,78].	- Deep-learning	- Probabilistic - Optimization - General	- CNN - SVM - Non-Maximum Suppression - Region Proposal Network - Multi Layer Perceptron - Maxpooling - Fast-RCNN - Transfer learning	- All the environment	- Consume a lot	- High	- Easy	- High

3. Mapping

Mapping is the task of finding a relevant representation of the environment's surroundings. This representation can be generated according to different criteria. For example, in autonomous driving systems, maps play an indispensable role in providing highly precise information about the surrounding of the vehicles, which aids in dealing with vehicles localization, vehicles control, motion planning, perception, and system management. Mapping also helps in better understanding of the environment. Maps seek information from different sensors, which are divided into three groups. The first one employs the Global Navigation Satellite System (GNSS) sensors with an HD map that relies on layers on a Geographic Information System (GIS) [18] (GIS is a framework that provides the possibility of analyzing spatial and geographic data). The second group is based on the range and vision-based sensors such as LiDAR, cameras, RADAR, etc., which help in the creation of a point cloud map. The third group uses cooperative approaches. Each vehicle will generate its local map and then assemble them through Vehicle-to-Vehicle (V2V) communication to generate the global map, which is lower in cost and more flexible and robust. Data received from these sensors are classified into stable objects in time, i.e., immovable, such as buildings, traffic lights, bridges, curbs, pavements, traffic signs, lanes, poles-landmarks. This information is tremendously essential to change lines, avoid obstacles, respect road traffic in general. Furthermore, the dynamic objects change coordinates over time, like vehicles, pedestrians, cyclists. This information is helpful in the context of using V2V or V2I communications. Other than that, it is not recommended to be mapped. Wong et al. [86] added another class named temporary objects, which are features that exist within a time period sense, like parked vehicles, temporary road works, and traffic cones. Maps are prone to be inaccurate due to errors of sensor measurements, hence, that can lead to failures in the positioning of objects on the map, which makes driving impossible in this situation. 10 cm of accuracy is recommended by Seif and Hu [87]. Moreover, the massive amount of data received from different sensors need high storage space and computational processing units. Fortunately, this can be covered by the new generation of computational capacities like NVIDIA cards. Ref. [88] discussed the required criteria for getting a robust map with respect to three aspects: storage efficiency, usability, and centimeter-level accuracy.

Our literature review found that maps for autonomous driving systems can be divided into two categories: offline maps and online maps. We will discuss each type of maps in more detail below.

3.1. Offline Maps

Offline maps (also called a priori or pre-built maps) are generated in advance to help autonomous driving systems in the navigation and localization tasks. According to [89], the turnover resulting from their usage increased to 1.4\$ billion in 2021 and is intended to reach 16.9\$ billion by 2030. The number of investments in this field tends to grow daily, and the competition between map manufacturers is at a high level. These companies use different technologies and strategies of partnership and collaboration. In the last few years, major auto manufacturers have accelerated the efforts to produce new generations of advanced driver assistance systems (ADAS). In consequence, developing such great maps is a must for these vehicles.

According to [90], three types of maps can be identified, including digital maps, enhanced digital maps, (HD) high-definition maps. Digital maps (cartography) are topometric maps that encode street elements and depict major road structures, such as Google maps, OpenStreetMap, etc. These maps also provide the possibility of finding the distance from one place to another. However, these maps are useless for autonomous vehicles due to the limited information provided, the low update frequency, and lack of connectivity. So, it cannot be accessed from external devices. The enhanced digital map is a little bit more developed than the first one. Furthermore, from what we identified, these maps are characterized by additional data information, including poles-landmarks, road curvature, lane level, metal barrier. The most important one is the HD map, which is a concept developed

by the Mercedes-Benz research planning workshop [91]. This map consists of a 3D representation of the environment which is broken down into five layers, i.e., Base map layer, Geometric map layer, Semantic map layer, Map priors layer, Real-time knowledge layer, as are clearly explained in [90,92].

To build a HD map, we should take into consideration four main principles mentioned in [90,92]. First, mapping a precomputation that facilitates the work of the autonomous driving system in real-time by solving, partially or completely, some problems in the offline stage in a highly accurate manner. The second principle is mapping to improve safety, highlighting the importance of providing accurate information about the surrounding objects to ensure safety, especially if we are talking about level four and five autonomous vehicles, which need more maintenance and surveillance. Alternatively, a map is another tool used at the moment of driving. The third principle is that a map should be robust enough against the dynamic objects that cause run-time trouble. Finally, a map is a global shared state working as a team, where each (AV) system provides its information. So, this idea can solve the problems related to the large computational memory usage as well as and reduce energy. Maps should be underpinned with hardware and software components. As aforementioned, hardware components are the data source of the map, while the software components are intended to analyze and manipulate, and even power the hardware [93].

Maps are prone to errors from various sources, including the change of reality, localization error, inaccurate map, map update error, etc., and we can define an accuracy metric to measure how much the map is an accurate representation of the real-world. According to [94], two metric accuracies are defined:

- **Global accuracy:** (Absolute accuracy) position of features according to the face of the earth.
- **Local accuracy:** (Relative accuracy) position of features according to the surrounding elements on the road.

In Table 3, we detail some companies that are developing digital map solutions for autonomous systems.

However, these maps are usually not publicly available and they need to be generated frequently, which give rise to the need for an in-house solution, and which can be created with vehicle sensors and with the help of academic solutions. So, all feature extraction algorithms we have already discussed are candidates for solving localization and mapping tasks. Features that are widely used for the purpose of creating a map are lanes [21,22,36], buildings [22,28,33], curbs [24,25,33], poles [26–30,32,33], or a combination of these. These maps are referred to as feature maps. There is a lot of types of maps in the literature, like the probabilistic occupancy map, which is a lightweight, and low-cost map that gives the probability that a cell is occupied or not, such as the work in [31]. Voxel grid map is another type of maps based on 3D voxelization of the environment; each voxel is equivalent to a pixel in a 2D environment. Also, these maps are used most of the time to discretize 3D points cloud, which reduces the work into voxels and offers more opportunities to employ relevant algorithms [76]. Schreier [95] adds more map representations depending on various levels of abstractness including parametric free space maps, interval maps, elevation maps, stixel world, multi-level surface maps, raw sensor data. Note that it is possible to combine the information from maps to gain a better understanding of the environment scene.

3.2. Online Maps

The localization and mapping problems were first treated separately, and good results were reported. In recent years, a significant advancement known as simultaneous localization and mapping (SLAM) has emerged to address the issue of non-existing mapping in indoor environments, or if there is no access to the map of the environment. This problem is more difficult because we try to estimate the map and find the position of the vehicles at the same time. For this reason, we call it online maps. In reality, there are many unknown environments, especially for the in-house robots. So the robot creates its own map map [96].

Let us look from the mathematics perspective, mainly from a probability and optimization viewpoint. We define the motion control by u_k (vehicle motion information using IMU, or wheel odometry, etc.), the measurement by z_k (providing a description of the environment using LiDAR, camera, etc.) and the state by x_k (the vehicle position). So we can distinguish between two main forms of SLAM problems, with the same importance.

The first one is the online SLAM which estimates only the pose at the time k and the map m . Mathematically, we search:

$$b(x_k, m) = p(x_k, m \mid z_{0:k}, u_{0:k}) \quad \text{which is called "belief"}$$

The second type is the full SLAM, which approximates the entire trajectory $x_{0:k}$ and the map m , expressed by:

$$b(x_{0:k}, m) = p(x_{0:k}, m \mid z_{0:k}, u_{0:k})$$

The online SLAM is just an integration of all past poses from the full SLAM:

$$\begin{aligned} p(x_t, m \mid z_{0:t}, u_{0:t}) \\ = \iint \cdots \int p(x_{1:t}, m \mid z_{0:t}, u_{0:t}) dx_0 dx_1 \dots dx_{t-1} \end{aligned}$$

The idea behind the online SLAM is to use the Markov assumption (i.e., the current position depends only on the last acquired measurement). With this assumption, SLAM's methods are also based on the Bayes theorem so that we can write [96,97]:

$$\text{posterior} = \frac{\text{likelihood} \cdot \text{prior}}{\text{marginal_likelihood}}$$

Then, employing this theorem to the belief $p(x_k, m \mid z_{0:k}, u_{0:k})$ and incorporating each term (namely, posterior, likelihood, and marginal likelihood) with its precise definition, including pertinent details. This process is referred to as the **Measurement_Update**.

$$p(x_k, m \mid z_{0:k}, u_{0:k}) = \frac{p(z_k \mid x_k, m) p(x_k, m \mid z_{0:k-1}, u_{0:k})}{p(z_k \mid z_{0:k-1}, u_{0:k})} \quad (1)$$

where:

- **Likelihood** = $p(z_k \mid x_k, m)$ which gives the possibility of making an observation z_k when the vehicle position is x_k and the set of landmarks m are known. Called in the literature the **observation_model** [97].

- **Prior** = $p(x_k, m \mid z_{0:k-1}, u_{0:k})$

$$\begin{aligned} &= \int p(x_k \mid x_{k-1}, u_k) \\ &\times p(x_{k-1}, m \mid z_{0:k-1}, u_{0:k-1}) dx_{k-1} \\ &= \int p(x_k \mid x_{k-1}, u_k) \cdot b(x_{k-1}, m) dx_{k-1} \end{aligned}$$

This calculation of the prior is called the **Prediction_Update**, which represents the best possibility of the state x_k . The term $p(x_k \mid x_{k-1}, u_k)$ is called **motion_model** [14,16,97].

- **Marginal_likelihood** = $p(z_k \mid z_{0:k-1}, u_{0:k})$ is a normalization term which depends only on the measurement and it can be calculated as given below [97]:

$$p(z_k \mid z_{0:k-1}, u_{0:k})$$

$$= \int p(z_k | x_k, m) p(x_k, m | z_{0:k-1}, u_{0:k}) dx_k$$

At this moment, let us replace what we got in the calculation of the prior in the Equation (1) and explore that the Marginal_likelihood is only dependent on the z_k to deduce this formulation [14,16]:

$$\begin{aligned} p(x_k, m | z_{0:k}, u_{0:k}) &= \frac{p(z_k | x_k, m) p(x_k, m | z_{0:k-1}, u_{0:k})}{p(z_k | z_{0:k-1}, u_{0:k})} \\ &\propto p(z_k | x_k, m) p(x_k, m | z_{0:k-1}, u_{0:k}) \\ &\stackrel{\text{prior}}{=} p(z_k | x_k, m) \int p(x_k | x_{k-1}, u_k) \cdot b(x_{k-1}, m) \end{aligned}$$

Table 3. Commercial companies are developing digital mapping solutions for self-driving cars.

Map	Original Country	Description/Key Features
HERE	Netherlands	Here is one of the companies that provides HD maps solutions and promise its clients to ensure the safety and get the driver truth by providing relevant information to offer vehicles more option and to decide comfortably. Here uses machine learning to validate map data against the real-word in the real-time, this technology achieve around 10–20 cm of the accuracy in the identification of the vehicles and their surrounding. Moreover, this map contains three main layers Road Model, HD Lane Model, and HD Localisation Model [98,99].
Tomtom	Netherlands	Support many applications in an (ADAS) like Hans-free driving, advanced lane guidance, lane split, curves speed warning, etc. Also, provide a high precise vehicle positioning with an accuracy of 1 m or better compared to reality and 15 cm in the relative accuracy. Also, Tomtom take the advantage of using the RoadDNA where it converts the 3D points cloud into a 2D raster image which made it much easier to implement in-vehicles. Tomtom consist of three-layer Navigation data, Planning data, RoadDNA [100,101].
Sanborn	USA	Exploit the data received from different sensors Cameras, LiDAR to generate a high-precise 3D base-map. This map attain 7–10 cm of the absolute accuracy [102].
Ushr	USA/Japan	Ushr use only stereo camera imaging techniques to reduce the cost of acquiring data, and they use advanced machine vision and machine learning enable to achieve 90% of automation of data processing, also this map has mapped over 200,000 miles under 10 cm level of absolute accuracy [103].
NVIDIA	USA	NVIDIA map detect semantics road features and provides information about vehicles position with robustness and centimeter-level of accuracy. Also, NVIDIA offers the possibility to build and update a map using sensors available on the car [104].
Waymo	USA	Waymo affirmed that a map for (AV) contain much more information than the traditional one and deserve to have a continuous maintain and update, due to its complexity and the huge amount of data received. Waymo extract relevant features on the road from LiDAR sensor, which help to accurately find the vehicle position by matching the real-time features with the pre-built map features, this map attains a 10 cm of accuracy without using a GPS [99].
Zenrin	Japan	Zenrin is one of the leaders' companies in the creation of map solution in japan, founded in 1948 and has subsidiaries worldwide in Europe, America, and Asia. The company adopt the Original Equipment Manufacturers (OEM) for the automobile industry. Zenrin offer 3D digital maps containing Road networks, POI addresses, Image content, Pedestrian network, etc. [105].
NavInfo	China	Explore the fusion of three modules HD-GNSS, DR engine, and Map fusion to provide high accurate positioning results. Also, they allow parking the vehicle by a one-click parking system. NavInfo is based on the self-developed HD map engine and provide a scenarios library based on HD map, simulation test platform, and other techniques [106].
Lvl5	USA	In lvl5, they believe that vehicles did not need to have a LiDAR sensor, unlike Waymo. The company The company focus on cameras sensor and computer vision algorithm to analyze and manipulate videos captured and converted into a 3D map. The HD maps created change multiple times in a day which is a big advantage compared with other companies. lvl5 get an absolute accuracy in the range of 5–50 cm [86,107].
Atlatec	Germany	To obtain a highly accurate map, Atlatec uses cameras sensors to collect data. After that, a loop closure is applied to get a consistent result. They use a combination of artificial intelligent (AI) and manual work to get a detailed map of the road objects. This map arrived to 5 cm in the relative accuracy [91,108].

The problem of SLAM, as we have noticed above, can be solved by following the two steps: **Measurement_Update** and **Prediction_Update**. So any solution to this problem

should be a great representation of the **observation_model** and the **motion_model**, which will give us the best estimation of the two steps aforementioned. The above explanation was about the probabilistic representation of the SLAM problem [16]. Now, let us change the focus to optimization and formalize the problem. Indeed, optimization is involved in resolving SLAM problems by minimizing a cost function which depends on the pose x_k and the map m with constraints. In the case of Graph SLAM, the cost function can be [14,109]:

$$Fun(x) = \sum_{ij} e_{ij}(x_i, x_j)^T \Omega_{ij} e_{ij}(x_i, x_j) \quad (2)$$

$$x^* = \underset{x}{\operatorname{argmin}} Fun(x)$$

where $x = \{x_0, x_1, \dots, x_i, x_j, \dots, x_n\}$ is the set of vehicles poses, m is the set of landmarks that consist the map, $e_{ij}(x_i, x_j) = z_{ij} - z_{ij}^*(x_i, x_j)$ is the error function which calculates the distance between the prediction and real observation in the node i, j , Ω_{ij} is the information matrix.

3.3. Challenges

- Data storage is a big challenge for AVs. We need to find the minimum information required to run the localization algorithm.
- Maps are updated frequently, so an update system should take place here to update and maintain the changes. Also, the vehicle connectivity system should support this task and supply the changes for other vehicles.
- Preserving the information on vehicle localization and ensuring privacy is also a challenge.

4. Localization

Localization is a crucial task within the context of any autonomous system development. This task is indispensable, i.e., to tell the vehicle where it is at each moment in time. Without this information, the vehicles can not avoid collision properly and can not drive in the correct line. At this level, we have seen in Section 2 how to extract relevant features from sensor measurements, and we have investigated different approaches to find the best representation of the environment (Section 3), which means that we have discussed the steps needed before localizing the vehicle. In the remainder of this section, we will depict what is new on localization approaches.

4.1. Dead Reckoning

Dead Reckoning (DR) is one of the oldest methods that determines the position of an object based-on three known pieces of information, including the courses that have been traveled, the distance covered (which can be calculated by using the speed of the object in the trajectory and the time spent within), and lastly a reference point, which is to know the departure point and the estimated drift, if there is. This method does not use any celestial reference, and it is helpful in marine navigation, air navigation, or terrestrial navigation [110,111]. So the coordinates can be found mathematically by:

$$x_k = x_0 + \sum_{i=0}^{k-1} d_i \times \cos \theta_i$$

$$y_k = y_0 + \sum_{i=0}^{k-1} d_i \times \sin \theta_i$$

where (x_0, y_0) is an initial position and d_i, θ_i are, respectively, the shortest way and the angle between the current and last position.

4.2. Triangulation

Triangulation strives to estimate vehicles' position based on geometrical properties. The principal idea is that if we have two known points, we can deduce the coordinate of the third point. For an accurate estimation of vehicles' positions, we should use at least three satellites. To solve this estimation, the procedure requires two steps as detailed below [110,112].

- **Distance estimation:**

This is approximately the distance between the transmitter (satellites) and the receiver (vehicle). In the literature, many methods have been implemented to estimate this distance. One of them is based on the Time of Arrival (ToA), which is the time that takes a signal to arrive at the receiver, and it is based on the signal propagation speed. So, the distance can be estimated (distance is equal to speed * time). One of the problems here is that the time should be synchronized between the sender and the receiver, which is difficult. An extension of this method is the Time Difference of Arrival (TDoA), which calculates the difference of time between two signals with different propagation speeds, from which we can deduce the position. There are other methods to do that, including Received Signal Strength Indicator (RSSI), Hop-based, Signal attenuation, Interferometry [110,112].

- **Position estimation:**

The position of the vehicles can be found based on the geometric properties and based on the distance estimated above. The calculation complexity depends on the dimensionality sought (2D or 3D) and the type of coordinates (Cartesian or spherical). Let us take a small example in 2D Cartesian coordinate system; see Figure 4 [113].

Based on Pythagoras's theorem:

$$r_1^2 = x^2 + y^2$$

$$r_2^2 = (U - x)^2 + y^2$$

Thus U is the first coordinates of C_2 ; see Figure 4.

$$x = \frac{r_1^2 - r_2^2 + U^2}{2U}$$

$$y = \pm \sqrt{r_1^2 - x^2}$$

We conclude that the remaining coordinates (x, y) can be found based on the distance to the satellites.

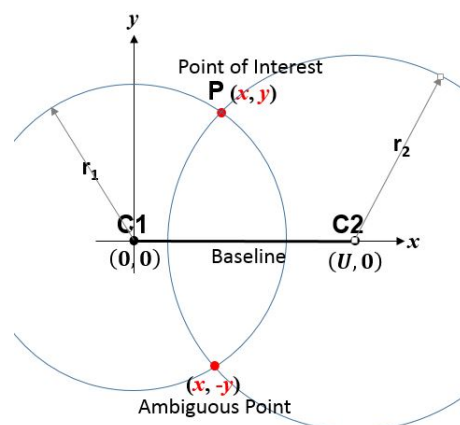


Figure 4. Cartesian 2D example scenario. C_1 and C_2 are the centers of circles (satellites). P is point of interest (vehicle) with coordinates (x, y) and r_1, r_2 are the distances from the center of circle, respectively from C_1 and C_2 , to the desired point [113].

4.3. Motion Sensors

Three principal components, namely acceleration, gyroscope, magnetometer, and motion sensor are used to detect object motions. Motions can be a translation over the axis x , y , or z , calculated by integrating the acceleration and computing the difference between the current and last positions. Motion can also be a rotation, which can be calculated by the gyroscope that provides the orientation using the roll, pitch, and yaw [110]. Much research has been done to use the motion sensors like IMU and wheel odometry to localize vehicles, and exploring machine learning algorithms in this field has shown its performance. Various architecture have been proposed. Input Delay Neural Network (IDNN) was used by Noureldin et al. [114] to learn error patterns in GPS outages. A comparative study was performed in [115], where they attempted to find the best guess of the INS during GPS outage, by testing different methods including Radial Basis Function Neural Network (RBFNN), propagation neural network, Higher-Order Neural Networks (HONN), Full Counter Propagation Neural network (Full CPN), back-propagation neural network, Adaptive Resonance Theory- Counter Propagation Neural network (ART-CPN), and the IDNN. Recent work by Dai et al. [116] used Recurrent Neural Networks (RNN) to learn the INS drift error, which is more robust and performs well given that the RNN is intended to solve time series problems. Onyekpe et al. [117] suggested that a slight change in the diameter of the tire or the pressure can lead to a miss-displacement in the odometry. Also, LSTM-based methods have been performed to learn the uncertainties in the wheel speed measurement that appear while the vehicle displacement is more potent than the INS solutions. The same authors extended their work to obtain computational efficiency and robustness against different GPS outages. The Wheel Odometry Neural Network (WhONet) framework justifies the use of the RNN network by examining various networks, including IDNN, GRU, RNN, and LSTM [118].

4.4. Matching Data

It is the task of matching data recorded from a sensor like signals, point cloud, and images with the real-world data reference to extract the autonomous vehicle's location. Some preliminaries are presented below to enable further discussions in this subsection. The Rigid transform matrix is in this form:

$$\left(\begin{array}{ccc|c} & & & t_x \\ & \mathbf{R} & & t_y \\ & & & t_z \\ ine0 & 0 & 0 & 1 \end{array} \right)$$

where R is the Rotation matrix, which is an orthogonal matrix with a determinant of ± 1 (i.e., $RR^T = R^T R = I_3$) and $T = (t_x, t_y, t_z)$ the translation vector. In the rest of this subsection we will present some approaches that are based on the matching process.

4.4.1. Fingerprint

This method enables the localization of a vehicle based on matching techniques. Indeed, the process is to create a database based on a reference car equipped with different sensors to record various environmental changes. After that, a "query" car will travel the environment and register its fingerprint, which will be compared with the database and pick the position that has the best match [110,119].

4.4.2. Point Cloud Matching

3D point cloud registration provides a highly accurate source of information that enables it to be employed in various domains, including surface reconstruction, 3D object recognition, and, lastly, localization and mapping in self-driving cars, which is our purpose. To do so, scan registration can present sufficient information for vehicle localization by aligning multiple scans in a unique coordinate system and matching similar parts from the

scans. In literature, several methods have been proposed to solve the scan matching. The Iterative Closest Point (ICP) is one of the earliest, proposed in 1992 by Besl and Mckay [120]. Its precision enables it to be very usable simultaneously in the positioning and mapping tasks. The idea behind this is to find a rotation matrix R and a translation vector T that will align the query scan with the reference scan iteratively. In fact, the aim is to minimize the distance between the points a_i from the target scan and their neighbor one b_i from the query scan [121]. M is number of point cloud in a scan:

$$\arg \min_{R,T} \left\{ \frac{1}{M} \sum_{j=1}^M \|a_j - (Rb_j + t)\|_2 \right\}$$

Note that, in each iteration, we stock the new value of the query scan $b_i' = Rb_j + T$.

However, this method suffers from some serious limitations and assumptions that can affect the convergence of the algorithm, including the good initialization of the algorithm, e.g., one surface being a subset of another, high computational cost, etc. [122]. As a consequence of these issues, many types of methods have been proposed to improve the ICP algorithm. The article [122] divides these extensions into five stages, depending mainly on the selection of the sampling and the feature metric used like point-to-point or point-to-plan methods, which effectively reduce the number of iterations. In recent years, the Normal Distribution Transformation (NDT) [123] emerged to solve the problem of measurement error and cover the problem of the no-correspondence relationship. Indeed, the idea is to break down the set of points' cloud into a 3D Voxels representation and assign for each voxel a probability distribution. So, even if we have a millimeter of error far from our real measurements, the NDT algorithm takes place to match far points based on the probability distribution [124]. As a result, some researches have combined the NDT process with the standard ICP algorithm, like in [125]. Another interesting approach, the Robust Point Matching (RPM), has been performed to overcome the problem of the initialization (RPM) and change the traditional correspondence into a 0, 1 matrix. Let m_i be the i_{th} point in M and s_j be the j_{th} point in S , so we define the correspondence matrix μ by:

$$\mu_{ij} = \begin{cases} 1 & \text{if point } m_i \text{ corresponds to point } s_j \\ 0 & \text{otherwise} \end{cases}$$

Thus, according to this method, the problem is to find the Affine transformation \mathbf{T} and the match matrix μ that make the best fit. The decomposition can give the transformation:

$$\mathbf{T}(m) = \mathbf{A}m + T$$

T is a translation vector, m is a point from the scan, \mathbf{A} is a matrix that depends on some parameter to be estimated based on a cost function [121,126]. An extension of this method is the thin plate spline robust point matching (TPS-RPM) algorithm, which augments the capability to handle the non-rigid registration [127].

Gaussian mixture models (GMM) family takes its place in this field as evidenced in literature. GMMReg [128] is one of them, where they try to represent the two points cloud into a GMM model to be robust to noise and outliers. Instead of aligning points, which is prone to error, they align distributions. e.g taking the intersection of these two distributions. Then, a standard minimization of the euclidean distance of the two new points cloud was performed to refine the transformation. DeepGMR [129] utilised the strategy of using deep learning methods to learn the correspondence between the GMM component and the transformation. Other methods based on deep learning can be found, like relativeNet [130], PointNetLK [131], Deep closest point (DCP) [132]. A complete survey on point cloud registration methods can be found in [133].

4.4.3. Image Matching

Images contain plenty of information that can be explored to do many things, like object tracking, segmentation, object detection, and so forth. Due to that, the vision-based localization is supported by geospatial information. It is now possible to estimate the poses of vehicles by checking the movement of a point (pixel or features) from one image to another. In the literature, vision-based localization is divided into local methods (or indirect or feature-based) and global methods (or direct or appearance or featureless) [134]. Local ones consider just a part of the local region that characterizes the images in the database and the query image. The global method, or the direct method, uses the movement of the intensity of each pixel without using a pre-processing step, such as feature extraction.

Based on [135], image-based localization can be divided into three groups. Primarily, the video surveillance based-methods track the vehicles based on cameras mounted on road infrastructure, and they determine their position based on cameras calibration after detection. Secondly, they tend to search the similarity between the query image and a pre-built database. So the process is called **similarity search**. After performing the feature extraction (Section 2), we receive a massive amount of data features that should be reduced to search quickly the similarity in images. Typically, we need to profit from these features as much as possible without losing the generalization. Quantization is one solution for that. This is the process of creating an index, which make it easy to search image by query. It adopt the concept of text research where they express each image by a vector of features, and a dictionary is built based on a large set of features which are extracted from visual documents, then applying a clustering method to reduce the size of the dictionary, each cluster centroid being called a visual word. Hence we can calculate the frequency of a specific visual word in the visual document, which is called the bag of feature (BOF) [136]. Another method called k nearest visual words was used in [137].

The similarity can be obtained by calculating the L2 norm between the identified features. However, this method does not work well in when there are of many descriptors, which means we need to find a robust algorithm that will search for similarity efficiently. Researches like [138,139] turn this problem into an SVM classification task. Ref. [140] presents a new architecture to associate the similarity features, called Multi-Task Learning (MTL). To improve the similarity process, we add another step of ranking, which is the task of underpinning the results of the retrieved data by classifying them into a list of candidates. For further information, the reader is referred to [8].

The last group is the visual odometry (VO) based methods, where the idea is to track the movement between two consecutive frames by matching the overlapping area. This method is widely used in the robotics field.

Let M_1, M_2 two image matrices, respectively, of image 1 and 2. The problem is to find a vector $t = [u, v]$ where:

$$\begin{bmatrix} p_{m_1} \\ p_{n_1} \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} p_{m_2} \\ p_{n_2} \end{bmatrix}$$

(p_{m_1}, p_{n_1}) and (p_{m_2}, p_{n_2}) are points, respectively, from image 1 and 2. The position of the vehicle can be calculated by minimizing the photo-metric error. The advantage of this method is that it can be performed even in a low-texture environment. However, it suffers from a high computational cost.

In the literature, we found some cost functions that we can use to find the best transformation vector t , such as to minimize the sum of squared difference (SSD):

$$SSD(t) = \sum_{p_m, p_n} (M_1(p_m, p_n) - M_2(p_m + u, p_n + v))^2,$$

Or minimizing the sum of absolute differences (SAD):

$$SAD(t) = \sum_{p_m, p_n} |M_1(p_m, p_n) - M_2(p_m + u, p_n + v)|,$$

Alternatively, maximizing the sum of cross-correlation coefficients.

$$CC(t) = - \sum_{p_m, p_n} M_1(p_m, p_n) M_2(p_m + u, p_n + v)$$

In the case of changing in the environment condition (i.e illumination), we must normalize another metric:

$$\begin{aligned} A &= M_1(p_m, p_n) - \bar{M}_1 \\ B &= M_2(p_m + u, p_n + v) - \bar{M}_2 \\ NCC(t) &= \frac{- \sum_{p_m, p_n} A * B}{\sqrt{\sum_{p_m, p_n} (A)^2 \sum_{p_m, p_n} (B)^2}} \end{aligned}$$

where \bar{M}_1, \bar{M}_2 are the mean intensities in each image. More information can be found in [134]. Moreover, the survey [8] divides the set of direct methods into three approaches. First class is Direct vision-based localization (VBL) with prior, which are a group of methods that are built under the assumption of an existing prior (from GPS, magnetic compass). From the methods that hold this assumption, we found the article [141], where they use an initialization from the GPS and compass embedded in a smart-phone to refine the global GPS coordinates. A coarse GPS initialization was given in [142] to perform a particle filter localization, also [143] used an indirect method as pre-processing step to refine estimation pose. The second class of methods is features (from images) to points (point cloud) matching, which find coherence between 2D image features and 3D points cloud, like in [144]. The third class is Pose regression approaches, which learn to regress the visual input data to the corresponding pose using mainly the regression forest method and CNN, like in [145,146].

4.5. Optimization-Based Approches

Optimization is the task of finding the best guess that can minimize or maximize a cost function with respect to defined constraints. These problems can exist in various domains, including operations research, economics, computer science, and engineering. In our case, we will use it to solve the localization and mapping problem, particularly the SLAM problem. Solving the SLAM problem is divided into front-end and back-end steps, from an optimization perspective. Front-end is the task of processing the data by extracting features, data association, checking loop closure, etc., in order to build a highly accurate map. The back-end step is responsible for finding the best guess of the locations of the vehicles by minimizing (or maximizing) a cost function [147].

One question is raised here: what is the difference between the loop closure and the re-localization. **Loop closure** is performed to cover the problem of error of sensors' measurement, because a measurement taken from any sensor is not what it is in reality. Also, the vehicles are prone to having different issues like drifting, acceleration changes, and weather conditions, which can affect the reliability and credibility of these measurements. Loop closure presents an aid in detecting whether the vehicle can re-visit the same location. If this were the case, the loop will be closed. See the example in Figure 5.

Re-localization is a task performed if the system fails to detect its position in the map. Often, this problem appears when there is an inadequate matching process. Thus we 're-localize' through place recognition.

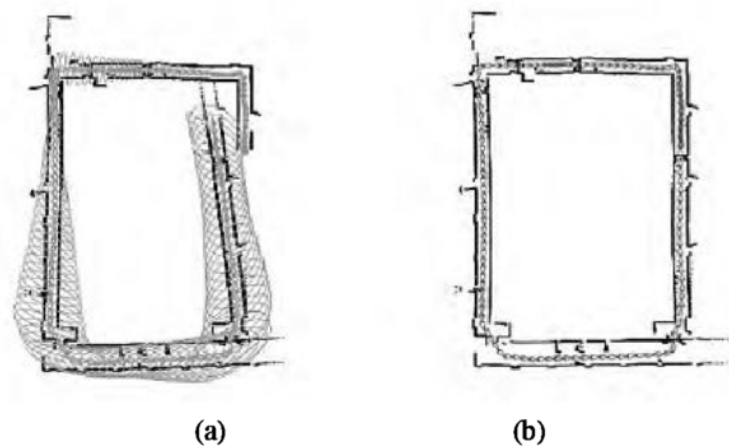


Figure 5. In a loop closure system, the recognition of a return to a previously visited location enables the correction of any accumulated errors in the system’s map or position estimate. Figure (a) shows an example before applying the loop closure. Figure (b) shows an example after applying the loop closure [148].

4.5.1. Bundle Adjustment Based Methods

One of the earliest methods that adopt the optimization techniques is the bundle adjustment. Bundle adjustment estimate simultaneously the camera pose parameter and the 3D point by minimizing a cost function. This method was used primarily in the 3D reconstruction. However, an adaptation was performed to apply this method to solve the SLAM problem. Due to its high performance, the Levenberg-Marquardt algorithm has been widely used to solve the optimization task [14,149]. Ref. [149] divides the series of bundle adjustment methods into a group that tries to improve the efficiency of the bundle adjustment algorithms and a second group that aims to reduce the frequency of invocation of individual bundle adjustment. The term bundle adjustment refers to the light ray that gets out from the features and converges to the camera center ‘bundle’, which should be ‘adjusted’ according to the features and camera center. The core idea is the sum of errors between 2D observations and the 2D points extracted from the re-projection from the 3D object with the help of camera parameters. Mathematically, the bundle adjustment is a non-linear least square problem:

$$\min \sum_{i=1}^n \sum_{j=1}^m (u_{ij} - \pi(C_j, X_i))^2$$

u_{ij} is what we observe in the pixel at i_{th} and j_{th} node, $\pi(C_j, X_i)$ is the operation that takes in as input the 3D points $X = \{X_i\}$ and the camera parameter $C = \{C_i\}$, which can be divided into intrinsic and extrinsic parameters. The output is the corresponding re-projected 2D coordinates. The process of this algorithm starts with the transformation from the world coordinates to the camera coordinates. Then, the 3D point coordinates are projected into a normalized image plane using triangular theory. After that, a distortion correction is applied. Finally, the remaining 2D point is deduced by projecting the corrected points into the pixel plan (intrinsic parameters) [149].

Attention-SLAM [150] is a newest method that uses a modified bundle adjustment (weight-BA) method in SLAM to reduce the trajectory error. Also, SalNavNet [150] was proposed to extract relevant features to help, as much as possible, to keep the vehicle on the trajectory. The method was tested in a dataset collected by the same authors named Euroc [151]. The test shows that the proposed method outperforms the ORB-SLAM method [152]. BALM [153] is a BA-based framework for LiDAR mapping. This method uses a new voxel architecture to adapt the use of BA in LiDAR. After that, the new BA method was incorporated with LOAM [154] to refine the map. Ref. [155] attempted to imitate the

human vision character by introducing a Salient Bundle Adjustment (SBA). DO-SLAM was proposed by Lu et al. [156]. They tried to combine direct-based and features-based methods to create an accurate SLAM framework. Indeed, the direct method was used for camera tracking, and the features method was used to infer the keyframe poses. The motion-only bundle adjustment gave an initialization pose.

4.5.2. Graph Based Methods

Recently, the SLAM problem has been solved by a new and fresh intuitive approach, the graph-based approach. This method involves building a graph whose nodes are the vehicles poses, and a measurement acquired and its edges can be considered as constraints. These constraints are obtained by the collected measurement and the motion actions. Once the graph is created (front-end), solving a minimization problem that involves finding the optimal configuration of the poses is required (back-end). For the first time, this method has suffered from the high computational time, which made it inapplicable, but with the development in the field of linear algebra, this method takes its place in the state-of-art of SLAM problem solver [157,158].

In regards to the mathematical approach, we focus on minimizing a specific cost function which is explained in depth in Section 3.2:

$$\begin{aligned} Fun(x) &= \sum_{ij} e_{ij}(x_i, x_j)^T \Omega_{ij} e_{ij}(x_i, x_j) \\ x^* &= \underset{x}{\operatorname{argmin}} Fun(x) \end{aligned} \quad (3)$$

where

$$e_{ij}(x_i, x_j) = z_{ij} - z_{ij}^*(x_i, x_j)$$

is the error distance between the real observation z_{ij} and the expected observation z_{ij}^* , and for the reason of abbreviation we will note it as e_{ij} . We will depict a traditional nonlinear least-squares optimization solution for this problem. Lets begin with applying first-order Taylor expansion to approximate the error e_{ij} .

$$\begin{aligned} e_{ij}(\check{x}_i + \Delta x_i, \check{x}_j + \Delta x_j) &= e_{ij}(\check{x} + \Delta x) \\ &\simeq e_{ij} + J_{ij} \Delta x \end{aligned}$$

where \check{x} is a good initialization and J_{ij} is the Jacobian of $e_{ij} = e_{ij}(\check{x})$ computed on \check{x} , Δx is an increment that should be estimated, because the optimal solution is equal to the sum of the initial pose and the increment.

$$x^* = \check{x} + \Delta x$$

We deduce the formulation (at the node i, j) by replacing the new error estimated:

$$\begin{aligned} &Fun_{ij}(\check{x} + \Delta x) \\ &= e_{ij}(\check{x} + \Delta x)^T \Omega_{ij} e_{ij}(\check{x} + \Delta x) \\ &\approx (e_{ij} + J_{ij} \Delta x)^T \Omega_{ij} (e_{ij} + J_{ij} \Delta x) \\ &= e_{ij}^T \Omega_{ij} e_{ij} + 2e_{ij}^T \Omega_{ij} J_{ij} \Delta x + \Delta x^T J_{ij}^T \Omega_{ij} J_{ij} \Delta x \\ &= c_{ij} + 2b_{ij} \Delta x + \Delta x H_{ij} \Delta x \end{aligned}$$

where

$$\begin{cases} c_{ij} = e_{ij}^T \Omega_{ij} e_{ij} \\ b_{ij} = e_{ij}^T \Omega_{ij} J_{ij} \\ H_{ij} = J_{ij}^T \Omega_{ij} J_{ij} \end{cases}$$

We generalize over all the nodes i, j . So we can rewrite the cost function:

$$\begin{aligned} Fun(\check{x} + \Delta x) &= \sum_{ij} F_{ij}(\check{x} + \Delta x) \\ &\simeq \sum_{ij} c_{ij} + 2b_{ij}\Delta x + \Delta x^T H_{ij}\Delta x \\ &= c + 2b^T \Delta x + \Delta x^T H \Delta x \end{aligned} \quad (4)$$

where

$$\begin{cases} c = \sum_{ij} c_{ij} = \sum_{ij} Fun_{ij}(\check{x}) = Fun(\check{x}) \\ b = \sum_{ij} b_{ij} \\ H = \sum_{ij} H_{ij} \end{cases} \quad (5)$$

The result is a quadratic form with a linear shift. So the minimum (after setting the gradient of Fun to 0) is:

$$H\Delta x = -b$$

which is a linear system that can be resolved with QR decomposition, Cholesky factorization, Levenberg Marquardt, etc. This theory works with the assumption of an Euclidean space of x , which is not the case most of the time for the SLAM problem. To solve that, a common approach is to solve the problem in a local Euclidean space called manifold [157,159].

Recently EyeSAM [160] introduced a novel graph-based approach based on the iSAM algorithm to help the surgeon under challenging eyes operations, especially in intraocular microsurgery. This paper aims to generate a map of vasculature points and their positions in the camera by exploring a series of inputs video frames. Jung and Choi [161] use an HG-SLAM: Hierarchical Graph-Based SLAM in a (UAV) Unmanned Aerial Vehicle with the aim of inspecting a bridge. Ref. [162] used a sensor fusion method to generate nodes by using visual-inertial (VI) (for camera and IMU) and normal distribution transform (NDT) odometry (for LiDAR), then, Generalized iterative closest point (G-ICP) matching is applied to define the constraint of the optimization problem. Finally, a Graph-based method (iSAM) was applied to refine the (UAV) trajectory. In [163] a robust ICP modification was proposed to solve the problem of the mismatched scanned image, then a Graph-based method was performed to deduce the robot path. This paper [164] introduces a Dynamic Object Tracking SLAM (DyOb-SLAM) system that enhances the conventional Simultaneous Localization and Mapping (SLAM) process by considering dynamic objects. Utilizing neural networks and optical flow algorithms, DyOb-SLAM effectively distinguishes between static and dynamic elements, resulting in separate maps. The system's real-time dynamic object tracking capability and accurate performance make it a promising solution for various robotic applications. DROID-SLAM [165], a pioneering deep learning-based SLAM system by Zachary Teed and Jia Deng, introduces recurrent iterative updates for camera pose and depth estimation through Dense Bundle Adjustment. Its exceptional accuracy and resilience surpass prior methods, with fewer catastrophic failures in challenging scenarios. Notably versatile, it leverages monocular, stereo, or RGB-D inputs during testing, yielding even better performance. With open-source code available on GitHub, DROID-SLAM fosters collaboration and advancements in the visual SLAM field.

4.6. Probability-Based Approches

In the probabilistic approach, solving the localization and mapping problem refers to finding a best estimation of:

$$p(x_k, m \mid z_{0:k}, u_{0:k})$$

A probabilistic estimation was calculated based-on the Markov and Bayesian assumption in Section 3.2:

$$p(x_k, m \mid z_{0:k}, u_{0:k}) \propto p(z_k \mid x_k, m) \int p(x_k \mid x_{k-1}, u_k) \cdot b(x_{k-1}, m)$$

where:

- $b(x_{k-1}, m)$: is the belief on x_{k-1}
- $p(z_k \mid x_k, m)$: is the observation model
- $p(x_k \mid x_{k-1}, u_k)$: is the motion model

Finding the best guess of the state x_k is by applying the **Prediction_Update** stage. Practically, it can be modeled mathematically by a non-linear or linear function (ϵ_{k-1} is an iid noise model) [97]:

$$x_k = f_k(x_{k-1}, u_k, \epsilon_{k-1})$$

And the **Measurement_Update** stage, which can be modeled by [97] (n_k is an iid noise model):

$$z_k = h_k(x_k, u_k, n_k)$$

These two stages depend mainly of the good estimation of **motion_model** $p(x_k \mid x_{k-1}, u_k)$ and the **observation_model** $p(z_k \mid x_k, m)$.

The **motion_model** gives an idea about how the state change over time given the state before. **observation_model** answers the question of if I have a state x_k and a control input u_k what will be the measurement z_k ?

Note that these formulations are intended to solve the SLAM problem. However, we can solve the problem interchangeably, estimating the localization problem and deducing the map or vice versa. In the rest of this section, we will propose some methods to solve just the localization problem, whereas we can adapt these methods for solving the SLAM problem.

4.6.1. Parametric Methods—Kalman Filter Family

Kalman filter (KF) is a method that aims to predict the future state based on the previous state. Typically, KF estimates the model's parameters when it is delicate to estimate them directly. KF attempts to reduce the error and noise that could affect the model's accuracy and provide fast and efficient data processing since it uses just the last previous state. Mathematically, the idea behind the KF is to find the \hat{x} that will maximize the posterior $p(x_k, m \mid z_{0:k}, u_{0:k})$ by using some analytical solution. In this case by a linear function where the Prediction_Update is [97,163]:

$$x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + \epsilon_{k-1} = f_k(x_{k-1}, u_k, \epsilon_{k-1})$$

And the Measurement_Update:

$$z_k = H_k x_k + n_k = h_k(x_k, u_k, n_k)$$

So the prediction of the state, at the time t is [166]:

$$\hat{x}_k = F_{k-1}\hat{x}_{k-1} + G_{k-1}u_{k-1}$$

$$P_k = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}$$

After exploring the previous state by the prediction state and the co-variance matrix, a Kalman gain was introduced to update these two last formulations [166]:

$$K_k = P_k \left(H_k P_k H_k^T + R_k \right)^{-1}$$

$$\hat{x}_k = \hat{x}_k + K_k(y_k - H_k\hat{x}_k)$$

$$P_k = (I - K_k H_k)P_k$$

where:

- F_k : is a matrix which represent the effect of the state x_{k-1} on the parameters at time t
- Q_k : the covariance of the process noise
- G_k : is a matrix that gives the effect pf the control input on the parameters
- H_k : is the observation matrix that the state parameters to the measurement domain
- P_k : is the co-variance matrix that describe the the correlation between parameters
- ϵ_k : is the motions noise
- n_k : is the measurement noise

KF assumes the linearity, and the noise should be Gaussian, which is not the case in real-life systems. Consequently, many extensions inherited from the KF concept aim to fix these limitations. Beginning with the Extended Kalman Filter (EKF), which is, in fact, a linearization of a non-linear system by applying the Taylor expansion, then, directly KF is performed on the remaining system. However, EKF suffers from the high computational cost, especially the calculation of the Jacobian matrix, which can affect the accuracy results. Using Unscented Kalman Filter (UKF), we apply a non-linear transformation (the Unscented Transform (UT)) for each sigma sampling point which is more adopted and robust for the actual case. Robust Kalman Filters (RKF) also has been used for linear and non-linear systems. This method uses linear matrix inequality that provides an upper bound for the co-variance error matrix [163,166]. More extensions, Cubature Kalman Filter (CKF), ensemble Kalman filter (EnKF), adaptive Kalman filter (AKF), switching Kalman filter (SKF), fuzzy Kalman Filter, motor EKF (MEKF), and more can be found in [167]. Most of them are interested in reducing large number of variables, linearization, minimization of the sampling set.

4.6.2. Non-Parametric Methods—Particle Filter Family

Particle filter (PF) was introduced in the 1990s to overtake the problem of non-linearity and non-Gaussianity. PF is a great solution for many applications, such as wireless network [168]. PFs have registered significant progress, with a large number of extensions that will be discussed later. This method comes with a robust idea that can tackle any system without any prior assumption of linearity or Gaussianity. Again, let us take a look at the mathematics of it. PF uses the sequential Monte Carlo method in which to represent the posterior $p(x_k | z_{0:k}, u_{0:k})$ by a set $\{w_{0:k}^i, x_{0:k}^i\}_{i=1}^{N_s}$ of N_s sample and weights:

$$p(x_{0:k} | z_{0:k}, u_{0:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(x_{0:k} - x_{0:k}^i)$$

where $x_{0:k}^i$ is a set of possible realization of the trajectory, $w_{0:k}^i$ is a set of weights that give a degree of importance to the sample $x_{0:k}^i$ that verified:

$$\sum_{i=1}^{N_s} w_k^i = 1$$

The higher the weight, the more important is the sample. $\delta(a)$ is the Dirac function (0 everywhere except a).

This approximation of the integrals by a discrete method enables an approximate probability distributions without regarding if it is Gaussian or not. However, one question can be raised here which is how we can sample from an unknown distribution $p(x_{0:k} | z_{0:k}, u_{0:k})$? So the idea is to use importance sampling, which is a method that generates particles from some usual distribution to deduce the distribution in question. Projecting this idea in our case and letting $q(x_k | z_{0:k}, u_{0:k})$ a known distribution, in which we can draw samples

and verify (for the sake of abbreviation we will give formulates without motion control measurement $u_{0:k}$)

$$q(x_k | z_{0:k}) = q(x_k | x_{k-1}^i, z_k)q(x_{k-1} | z_{0:k-1}) \tag{6}$$

So the new approximation is:

$$p(x_k | z_{0:k}) \approx \sum_{i=1}^{N_s} \tilde{w}_k^i \delta(x_k - \tilde{x}_k^i)$$

where:

$$\tilde{x}_k^i \sim q(x_k | z_{1:k})(i = 1, 2, \dots, N_s)$$

and

$$\tilde{w}_k^i \propto \frac{p(x_k^i | z_{1:k})}{q(x_k^i | z_{1:k})}$$

Practically, \tilde{w}_k^i formulation should be a recursive one, i.e., depending on its last step \tilde{w}_{k-1}^i which is more suitable for calculation. (applying Bayesian theorem in the nominator and formulate (5) in the denominator) gives:

$$\begin{aligned} \tilde{w}_k^i &\propto \frac{p(x_k^i | z_{1:k})}{q(x_k^i | z_{1:k})} \\ &\propto \frac{p(z_k | x_k^i)p(x_k^i | z_{1:k-1})}{q(x_k^i | x_{k-1}^i, z_k)q(x_{k-1}^i | z_{1:k-1})} \\ &\propto \frac{p(z_k | x_k^i)p(x_k^i | z_{1:k-1})}{q(x_k^i | x_{k-1}^i, z_k)} \end{aligned} \tag{7}$$

On the other hand (demonstration is in [168])

$$p(x_k | z_{1:k-1}) = \sum_{i=1}^{N_s} \tilde{w}_{k-1}^i p(x_k | \tilde{x}_{k-1}^i)$$

So

$$p(x_k^i | z_{1:k-1}) = \tilde{w}_{k-1}^i p(x_k^i | \tilde{x}_{k-1}^i)$$

By now we replace in (6) and we get

$$\tilde{w}_k^i \propto \tilde{w}_{k-1}^i \frac{p(z_k | x_k^i)p(x_k^i | \tilde{x}_{k-1}^i)}{q(x_k^i | x_{k-1}^i, z_k)}$$

Common choice of the importance density in the robot localization in general is [97] $p(x_k^i | x_{k-1}^i)$ so

$$\tilde{w}_k^i \propto \tilde{w}_{k-1}^i p(z_k | x_k^i)$$

According to [97], PF is fronted by some serious challenge that could generate difficulties in the convergence including:

- Degeneracy Problem: which appears in the case of a one weight particle being closed to 1, whereas the other are close to be 0 due to the increase of the unconditional variance of the importance weights. This degeneracy can be measured by the effective sample size (ESS)

$$ESS = \frac{N_s}{1 + \text{var}(\tilde{w}_k^i)}$$

A big variance means a small ESS which lead to the degeneracy problem.

Hints: Re-sampling after the measurement update.

- Sample Impoverishment: inherent from the Degeneracy Problem, is where the particles collapse into a single point due to the low variance provided in the prediction update

Hints: Generate a sufficient amount of noise in the prediction update.

- Divergence: Caused by various things, including the inconsistent measurement, failures in the hardware, etc.

Hints: Prevent these causes by monitoring and update.

- Selecting the Importance Density: because it is the core of particles filter. Generating particles from a non-efficient Importance Density means that particles are out of range and will not converge to the state.

Hints: Choosing a big one that will maximize the signal to noise ratio (SNR) ratio [169].

- Execution: a large number of particles directly affecting the computational cost, a small number of particles affecting the convergence of the method.

Hints: Check an optimal amount.

Due to these challenges, PF has been widely extended and developed to cover the discussed issues. An interesting one is the Unscented Particle Filter (UPF). An unscented transform is taken here to approximate the transformation of Gaussian variables [168]. Auxiliary Particle Filters (APF) is another extension that aims to highlight the particles that have a high likelihood by reusing the last measurement in the prediction step [97]. Extended Kalman Particle Filter (EKPF) is a mix of KF and PF to construct a robust importance density that can boost the experiments' performance. Rao–Blackwellized Particle Filter (RBPF) is a very important improvement, where they divide the state $x_{0:k}$ into two chunks $x_{0:k}^1$ and $x_{0:k}^2$. Thus, the posterior $p(x_{1:k} | z_{1:k}) = p(x_{1:k}^1, x_{1:k}^2 | z_{1:k})$ can be written as follow

$$p(x_{1:k}^1, x_{1:k}^2 | z_{1:k}) = p(x_{1:k}^1 | z_{1:k}) \cdot p(x_{1:k}^2 | x_{1:k}^1, z_{1:k})$$

The core advantage of this method is the reduction in time consumption. Another method is by reducing the dimensionality by the use of Adaptive Particle Filter (AdPF), which is based on varying the number of particles [97].

4.6.3. Localization with Filter-Based Methods

PF and KF are widely used in the literature for localization vehicles like in [22], who used a KF to localize the vehicles based on the road marking features and LiDAR reflectivity. Ref. [25] used double-layer KF to refine the location after extracting curbs features. Ref. [26] utilized a particle-based localization framework. Ref. [28] made use of a PF layer and a histogram filter layer based on the vertical features and ground features. In [29,31], PF-based localization with some modification is also used. The SLAM problem, in the meanwhile, was very well treated with these filters. Invariant Kalman filter was proposed in [170] to solve the SLAM problem, which is based on an Unscented KF and the theory of lie group. Ref. [171] is an improvement of the UKF-SLAM, where a maximization method was designed to find the best parameters-collection for the model. A Rao–Blackwellized particle filter was used in [172] to solve the Monocular SLAM problem with a special weighting method. Zhang and Li [173] attempted to combine a probabilistic architecture with graph theory. Indeed, the authors proposed a Bathymetric Particle Filter mixed with graph-based theory. This study [174] offers a comprehensive analysis of LiDAR-based Simultaneous Localization and Mapping (SLAM) for autonomous vehicles. The authors employ an adaptive Kalman filter (KF) to fuse landmark sensor and IMU data, enhancing accuracy in real-time scenarios. Closed-form expressions for crucial SLAM components aid real-time implementation, and observability analysis demonstrates the system's reliability. This research advances LiDAR-based SLAM for autonomous vehicles by integrating IMU data through an adaptive Kalman filter, improving accuracy and real-time implementation while ensuring robust performance. Ref. [175] presents a novel approach to radio-based vehicular Simultaneous Localization and Mapping (SLAM) using a sequence of three

Poisson multi-Bernoulli mixture (PMBM) filters. These filters progressively enhance SLAM accuracy while reducing complexity, offering a promising solution for efficient vehicular SLAM in real-world scenarios. Ref. [176] offers a tutorial on SLAM Backends with Objects in Motion, introducing an extended optimization-based framework for dynamic scenes. The authors present a new algorithm, dynamic EKF SLAM, which efficiently handles moving objects and features while ensuring accurate localization, object pose estimation, and map generation. This tutorial provides valuable guidance for addressing dynamic challenges in SLAM applications, bridging recent advancements with practical solutions.

4.7. Cooperative Localization

A more recent, and accurate approach is cooperative localization. According to [177], three key elements should be provided for successful autonomous vehicles, first, fusing information collected by the sensors. Second, integrating a high precision map. Third, providing technology to ensure communication which is the subject of interest of this section. The core idea is to create a communication between the vehicles with something (lets call it X) in order to provide a fluid exchange of information, mainly their positions, speed, etc. This communication can be created using some devices, including WiFi, cellular, and UWB radio communications [18]. However, what does 'X' mean, and what are the exciting communications that exist? [178] (see Figure 6);

- **Vehicle-to-Vehicle (V2V):** generate such a network between the vehicles where the information can be transmitted; this information (position, speed, orientation, etc.) can be used interchangeably between vehicles to perform principal tasks like avoiding collision, changing lines.
- **Vehicle-to-Infrastructure (V2I):** create communication between vehicles based on an intermediate infrastructure that is suitable to share information, from the node to any connected vehicles.
- **Vehicle-to-people (V2P):** most generally uses the interaction of vehicles and other road users like walking people, people using mobile devices or bicycles or wheelchairs, etc.
- **Vehicle-to-Cloud (V2C):** this is a technology that enables the vehicles to connect to the cloud-based on a cloud-based connected platform and via broadband cellular connectivity.
- **Vehicle-to-X (V2X):** Offers the possibility of connecting the vehicles with everything, which is a great idea because providing information from road users' early gives enough time to execute, in a comfortable way, vehicles' task and checking the prioritization so that to increase the safety.

With regards to V2V mode of communication, V2V requires a wireless signal to ensure communication between vehicles. In fact, V2V works in a Vehicular Ad-Hoc Network (VANET) environment, which is a temporary network environment where vehicles communicate with each other [18,177]. V2V has so many applications, including emergency brake lights, pre-crash sensing, blind intersection warning, forward collision, left turn assist, stop sign assist, roadside service finder, etc. [177]. The history of information (like the position, speed, travel direction, acceleration) consists of the core messages that should be transmitted between vehicles to deal with the randomness of life road events. Most of the time, performing this communication requires the IEEE 802.11p protocol, which is an extension of the IEEE 802.11 protocol [18]. The paper [179] introduces the A-DRIVE protocol to address deadlocks in connected and automated vehicle scenarios. It consists of two components: V2V communication-based A-DRIVE and Local perception-based A-DRIVE. Simulations show improved traffic throughput and reduced recovery time compared to a baseline protocol. The protocol presents a promising solution for enhancing intersection efficiency and safety in various road situations. The paper [180] presents innovative distributed multimodal localization methods for connected and automated vehicles (CAVs) that use information dissemination through vehicular communication networks. The methods use Adapt-then-Combine strategies with Least-Mean-Squares (LMS) and Conjugate Gradient

(CG) algorithms to enable cooperative fusion of different measurement modalities, allowing better position estimation without centralized control or GPS. Simulation studies, using kinematic models and the CARLA simulator, validate the superiority of the proposed approaches over existing methods in terms of accuracy and convergence, and provide valuable information for the design of robust localization strategies in CAV systems. The study [181] introduces an optimization-based path-planning algorithm for both connected and non-connected automated vehicles on multilane motorways. The algorithm optimizes vehicle advancement, passenger comfort, collision avoidance, and other factors within a model predictive control framework. Numerical simulations demonstrate that optimally controlled automated vehicles exhibit improved speed adherence and traffic flow efficiency, with connected controlled vehicles showing even higher efficiency. The paper [182] introduces a fully distributed control system for coordinating connected and automated vehicles (CAVs) at intersections. The approach uses a model predictive control framework combined with V2V communication, localization, and collision estimation. Experimental tests demonstrate its efficacy in ensuring safe and efficient traffic flow, offering a promising solution for future transportation systems. The research [183] delves into the application of Infrastructure-to-Vehicle (I2V) data in autonomous driving under challenging visibility and dense traffic scenarios. The paper emphasizes the advantages of leveraging external sensors to enhance a vehicle's perception capabilities. However, the critical concern of cyber-security in I2V communication is addressed. To mitigate this, the authors propose an anomaly detection algorithm for elevating the vehicle's self-awareness at intersections. This algorithm, integrated into autonomous vehicles, evaluates I2V communication health against potential cyber attacks. When employing cyber-attack scenarios from the Secredas Project [183], the study uses a simulation environment to analyze the impact of I2V attacks on autonomous vehicle performance, particularly in situations where sensor redundancy is compromised. The findings reveal that the introduced anomalies during such attacks can be effectively identified and managed by the autonomous vehicle, ensuring safer navigation through intersections while maintaining precise object tracking.

Probabilistic approaches were very used with the cooperative localization. Ref. [184] used features like inactive cars, traffic lights, or people who are linked with V2V communication to be fed as an input to Kalman filter method based on GPS. Also, Ref. [185] proposed using the Enhanced Innovation-Based Adaptive Kalman Filter (EIAE-KF) architecture by fusing the Innovation-based Adaptive Estimation Kalman Filter (IAEKF) with vehicular kinematic information. One of the most recent approaches [186] outperformed the GPSIDR system and Extended Kalman Filter (EKF), which uses information (i.e., location, speed, acceleration, etc.) to feed the Unscented Kalman Filter (UKF). A particle-based method was used in [187] based on the roadside unit (RSU), which achieved an error of 0.1–0.2 m. A detailed survey was performed in [17]. This paper [188] introduces crowdsourced vehicle trajectories as a solution to enhance autonomous vehicle (AV) mapping and navigation in work zones. By incorporating Gaussian Mixture Models (GMM) and occupancy grid maps (OGM) based on human-driven data, the proposed approach improves AVs' understanding of drivable areas in changing environments. Compared to pure SLAM methods, the introduced crowdsourcing technique significantly enhances AVs' ability to accurately define drivable areas, ensuring better path planning and control. This research contributes to the advancement of AV technology in work zones, showcasing the potential of integrating crowd-sourced data for improved mapping and navigation in challenging scenarios. The authors of [189] propose a labeled multi-Bernoulli filter solution for connected vehicles, utilizing local consensus iterations for complementary fusion of sensor data. They introduce a novel label merging algorithm to address double counting and extend the label space for improved consistency. Experimental results demonstrate its superiority over standard approaches, enhancing situation awareness in connected vehicles for applications like autonomous driving and intelligent transportation systems.



Figure 6. Illustration of the V2V-V2I-V2P-V2X.

4.8. Discussion, Challenges, and Future Directions

We have surveyed in this section the approaches used to solve the problem of localization in self-driving vehicles (even for any autonomous systems in some cases). We have summarized in Table 4 those approaches, applicable sensors, their accuracy in real environment scenarios, and the main challenges that should be taken into account in the upcoming research. Using some of these approaches must ensure fast interaction with other vehicle's tasks, which reduces time and energy consumption for the vehicles and save battery life for the long term. Moreover, we need to develop a self-driving vehicle that can overcome the challenges presented by rapid and violent shaking environments. This vehicle must ensure the accuracy of the laser odometer and effectively manage the accumulation of errors during extended long-distance operations. We need also to ensure the reliability of the methods, and provide options to supply other vehicles with information, which creates the challenge of privacy and security in this application domain. Also, another interesting challenge is the conflict between information received from the sensors and what we got from the nearby connected vehicles. Furthermore, developing self-driving tech involves other crucial challenges like processing sensor data in real-time for split-second decisions, creating detailed semantic maps to understand road elements deeply, adapting to adverse conditions like heavy rain, and handling uncommon scenarios confidently. These advancements are key for safer and more reliable autonomous driving.

Table 4. Summary of existing approaches to solve the localization problem.

Approach	Sensors	Score	Main Challenges
Dead Reckoning	GPS INS Wheel odometry	Low	Prone to error in case drifting
Triangulation	GNSS	High	Limited in some environment scenarios
Motion tracking	LiDAR Cameras	Passable	High cost of time and energy A few objects in the environment mislead the matching process.
Optimization	LiDAR Cameras	High	High computational time to solve the matrix calculation
Probability	LiDAR Cameras	High	Problem of initialization
Cooperative localization	Vehicle connectivity	High	Limitation if there is a long distance between vehicles

5. Security in Localization

Integrating the revolution on telecommunication technologies in autonomous vehicles does bring about the problem of data security given the massive amount of data received from different sources (GPS, IMU, Cameras, LiDAR, etc.) and transmitted (or received) via different connection means (WiFi, Bluetooth, cellular connection, IEEE802.11p, WAVE). Attack threats, according to [190], could affect most of the sensors, and they can be performed remotely without using any attached component and without modifying parts of the AV. These attacks are not executed vainly, without a purpose or benefit. So, they should be interrupted from corrupting the autonomous system and stealing personal information, such as the location, destination, or other confidential information. In this article [191], the authors underscore the rising significance of cybersecurity within the realm of autonomous vehicles. As these vehicles increasingly depend on intricate embedded systems and external connectivity to process their surroundings and make informed decisions, the heightened connectivity exposes them to potential cyber threats of grave consequences. The authors highlight the existing trend of automotive system attacks and predict a surge in such threats with the proliferation of autonomous vehicles. Given these vulnerabilities, the urgency to fortify cybersecurity protocols becomes paramount. To tackle this challenge, the authors propose a comprehensive roadmap aimed at establishing secure autonomous vehicles. They delve into noteworthy past automotive cyber-attacks and present contemporary AI-driven solutions as a means to bolster cybersecurity. Leveraging AI technologies, such as machine learning and deep learning algorithms, proves effective in promptly detecting and mitigating potential cyber risks. Nevertheless, despite these advancements, there remain crucial open challenges. Among these are real-time threat identification and response capabilities, development of robust authentication mechanisms for vehicle-to-vehicle communication, and implementation of secure over-the-air software updates. Ultimately, safeguarding cybersecurity takes precedence as autonomous vehicles redefine transportation safety and convenience. The proposed roadmap delineates the integration of AI to elevate cybersecurity in autonomous vehicles, offering insights into the mitigation of challenges and implementation of robust security measures for a safer autonomous transportation future. According to [190], most of the defense methods can be summarized into the Anomaly-based Intrusion Detection System (IDS) category, which can help detect unauthorized access data. Also, the abundance of information can be obtained by gathering data from different AVs, which can then be used to increase confidentiality. The final category is the Encryption method, which is intended to help sensors that lack authentication like Controller Area Network (CANs). In Table 5, we surveyed some interesting attacks that threaten different sensors of the AVs. Also, we present some key solutions to prevent these attacks.

This study [192] introduces a cloud-computing-based approach to enhance the resilience of Connected and Automated Vehicles (CAVs) against potential malicious vehicle attacks. By sharing local sensor data and employing a cloud-based sensor fusion algorithm, the proposed method aims to identify and isolate malicious information while maintaining accurate state estimation for legitimate vehicles. Numerical examples demonstrate the effectiveness of this approach in detecting and addressing malicious vehicles, contributing to the security and reliability of CAVs in autonomous driving scenarios. The [193] paper introduces a novel approach, for secure vehicle communication. It utilizes secure clusters based on vehicular secrecy capacity, hash chains, and Mobile Edge Computing. The proposed method holds potential to enhance vehicle communication security and efficiency, contributing to the development of autonomous driving technologies. The authors of [194] establish a practical definition of a security cluster in vehicle-to-vehicle (V2V) communications based on secrecy capacity, specifically with respect to Signal-to-Noise Ratio (SNR) values. They incorporate conventional and vehicle-related parameters, such as vehicle speed and safety distance, to model secrecy capacity. This approach ensures real-time control of secrecy capacity, contributing to robust V2V communication security. Overall,

the paper advances the field of autonomous driving systems by enhancing communication security through a comprehensive approach that considers various parameters.

Table 5. A description of some of the attacks that threaten vehicle security systems and possible solutions.

Name	Devices	Access	Attack Effect	Countermeasure
Malicious OBD devices	OBD	Physical	<ul style="list-style-type: none"> Controlling some AV components Injecting codes to ECUs 	<ul style="list-style-type: none"> Authenticity, Integrity Reduce the time of execution [190]
CAN access through OBD	CAN	Physical	<ul style="list-style-type: none"> Eavesdrop CAN messages from the ODB Insert code in the ECUs 	<ul style="list-style-type: none"> Reinforce authenticity and confidentiality by proposing a robust methods of encryption [195] Detecting attempts to transmit messages to the ECUs [196]
ECU access through CAN in LiDAR spoofing	ECU	Physical	<ul style="list-style-type: none"> Change/add codes to the ECUs 	<ul style="list-style-type: none"> Same as the first two attacks
	LiDAR	Remote	<ul style="list-style-type: none"> ECUs make believe that there is an object close 	<ul style="list-style-type: none"> Signal filter, using machine learning [197]
LiDAR jamming	LiDAR	remote	<ul style="list-style-type: none"> Denial-of-Service 	<ul style="list-style-type: none"> Changing wave frequency [198] Integrating V2V communications [198] Multiple LiDAR views [198]
Radar spoofing	Radar	Remote	<ul style="list-style-type: none"> False distance estimation to the object 	<ul style="list-style-type: none"> [199] Propose (PyCAR) that detect malicious signals by transmitting a challenging signal that will be compared with certain threshold
Radar jamming	Radar	Remote	<ul style="list-style-type: none"> Miss detection object 	<ul style="list-style-type: none"> Distinguish the correct signal from the counterfeit one [190]
GPS Spoofing	GPS	Remote	<ul style="list-style-type: none"> False GPS location 	<ul style="list-style-type: none"> Controlling the GPS absolute, relative signal strength perform a time comparison Use redundant information from multiple-GPS satellites [190]
Camera blinding	Camera	Remote	<ul style="list-style-type: none"> Miss object detection 	<ul style="list-style-type: none"> Use multiple-cameras to increase the redundancy [200]
Adversarial Images	Images	Remote	<ul style="list-style-type: none"> Provide incorrect prediction 	<ul style="list-style-type: none"> Increase the performance of The pre-processing step after feeding to The machine learning algorithms
Falsified information	V2V & V2I	Remote	<ul style="list-style-type: none"> Threat the AVs operations and the flow traffics 	<ul style="list-style-type: none"> Reinforce the authentication schemes [190]

6. Environmental Impact of Localization and Mapping Techniques in Self-Driving Vehicles

In this section, we present a comprehensive exploration of the potential impact of autonomous driving localization and mapping techniques. Our research covers a range of factors, including energy consumption and emissions reduction, noise and light pollution, integration of autonomous vehicles into smart cities, optimization of infrastructure to support autonomous driving technology, and refinement of software algorithms to improve performance.

- Energy Consumption and Emission Reduction (air pollution):** Self-driving vehicles require energy-intensive sensors, computing systems, and data centers for mapping, potentially increasing overall energy demand. Ref. [201] affirmed that a connection exists (mutual causation) between energy usage and CO₂ emissions. Energy consumption influences CO₂ emissions (in the near term), indicating that higher energy consumption might result in increased CO₂ emissions, and conversely (in the extended period). The [202] paper mentioned that current autonomous electric vehicles, which are based on conventional propulsion systems, have limitations in terms of energy efficiency and power transmission. These limitations may hinder large-scale adoption in the future. To address this issue, a study was conducted to analyze the energy consumption and efficiency improvement of a medium-sized autonomous electric vehicle powered by wheel motors. The study involved the development of a numerical energy model, which was validated against real driving data and applied in a case study. The energy analysis focused on three driving conditions: flat road, uphill and downhill. This allowed the energy consumption to be examined and the potential for energy savings through the use of an in-wheel drive system.

The analysis took into account factors such as regenerative braking, which allows energy to be recovered during deceleration. Energy consumption and regenerated energy were calculated based on vehicle dynamics and autonomous driving patterns specific to each driving cycle. A case study was conducted based on driving data from electric vehicles in West Los Angeles. In this chapter [203], the authors introduce a novel framework aimed at creating energy-efficient autonomous driving policies for shared roads. This innovative approach combines cognitive hierarchy theory and reinforcement learning to address the complexities of human driver behavior. Cognitive hierarchy theory is employed to model decision-making at various levels of rationality exhibited by human drivers. By comprehending human choices, autonomous vehicles (AVs) enhance their ability to predict and adjust to these behaviors, facilitating safe navigation on shared roads. The framework also integrates reinforcement learning, enabling AVs to continually learn from their environment and enhance their decision-making skills. This iterative learning process empowers AVs to adapt to changing road conditions, optimize energy usage, and ensure consistent safe performance. These vehicles grapple with the dual challenge of precise sub-centimeter and millisecond self-localization while maintaining energy efficiency. In this context, localization takes center stage. This paper [204] presents FEEL—an inventive indoor localization system that amalgamates three low-energy sensors: IMU (Inertial Measurement Unit), UWB (Ultra-Wideband), and radar. The authors elaborate on FEEL's software and hardware architecture. The paper also introduces an Adaptive Sensing Algorithm (ASA) that optimizes energy usage by adapting sensing frequency according to environmental dynamics. By judiciously curtailing energy consumption, ASA achieves up to 20% energy savings with minimal impact on accuracy. Extensive performance evaluations across various scenarios underscore FEEL's exceptional accuracy, with deviations of under 7cm from ground truth measurements, coupled with ultra-low latency of around 3ms. These findings underscore FEEL's prowess in satisfying the stringent demands of AIV localization. The proposed control strategy in [205] considers energy consumption, mobility, and passenger comfort while enabling vehicles to pass signalized intersections without stops. The leader vehicle plans its trajectory using dynamic programming (DP), optimizing energy consumption and other factors. Following CAVs either use cooperative adaptive cruise control or plan their own trajectories based on intersection conditions. Simulation results demonstrate reduced energy consumption without compromising mobility.

- **Sound and light pollution:** Sound and luminous pollution are additional plausible environmental consequences associated with autonomous vehicle operation. The paper [206] aims to introduce the project “DICA-VE: Driving Information in a Connected and Autonomous Vehicle Environment: Impacts on Safety and Emissions”. It also proposes a comprehensive approach to evaluate driving behavior volatility and create alerts for mitigating road conflicts and noise emissions in a connected vehicle setting. While literature on autonomous driving often overlooks noise and light pollution, these issues carry significant consequences for ecosystems and communities. Quieter driving and reduced vehicle volume may alleviate noise pollution. Additionally, AVs driving in darker conditions could decrease artificial lighting. These considerations call for more attention [207].
- **Smart cities:** The idea of smart cities necessitates merging multiple technologies—immersive sensing, ubiquitous communication, robust computing, substantial storage, and high intelligence (SCCSI). These are pivotal for applications like public safety, autonomous driving, connected health, and smart living. Foreseeing the rise of advanced SCCSI-equipped autonomous vehicles in smart cities, [208] proposes a cost-efficient Vehicle as a Service (VaaS) model. In VaaS, SCCSI-capable vehicles act as mobile servers and communicators, delivering SCCSI services in smart cities. VaaS's potential in smart cities is explored, including upgrades from traditional vehicular ad-hoc networks (VANETs). The VaaS architecture effectively renders SCCSI services, addressing

architectural, service, incentive, security, and privacy aspects. The paper offers a sustainable approach using VaaS, shedding light on SCCSI integration in smart cities through autonomous vehicles for diverse applications. The research in [209] examines autonomous vehicles (AVs) as potential smart and sustainable urban transportation amidst rapid urbanization. Growing urban mobility needs necessitate sustainable solutions to counter adverse societal, economic, and environmental effects. Addressing privacy and cybersecurity risks is pivotal for AVs' development in such cities. The study evaluates global government measures to manage these risks. Privacy and cybersecurity are vital factors in AVs' smart and sustainable city integration. The authors review literature supporting AVs' role in sustainable development. Governments enforce regulations or guidelines for privacy. Cybersecurity relies on existing regulations while partnering with private sector for improvements. Efforts by countries like the US, UK, China, and more, along with state-level actions, tackle AV-related risks. This study offers a comprehensive analysis of AVs' privacy and cybersecurity implications in smart, sustainable cities. It underscores global governmental actions and their significance in ensuring safe AV deployment, providing insights crucial for future transport systems. The [210] paper introduces AutoDRIVE, a comprehensive research and education ecosystem for autonomous driving and smart city solutions. AutoDRIVE prototypes, simulates, and deploys cyber-physical solutions, offering both software and hardware-in-the-loop interfaces. It is modular, scalable, and supports various frameworks, accommodating single and multi-agent approaches. The paper showcases ecosystem capabilities through use-cases: autonomous parking, behavioral cloning, intersection traversal, and smart city management. AutoDRIVE validates hardware-software components in intelligent transportation systems. As an open-access platform, AutoDRIVE aids prototyping, simulation, and deployment in autonomous driving and smart cities. Its adaptability to diverse frameworks and expandability aligns with field trends. AutoDRIVE enhances research and education in autonomous driving effectively.

- **Infrastructure optimization:** The study in [211] explores integrating solar energy into mobility systems, particularly Solar-electric Autonomous Mobility-on-Demand (AMoD). Solar roofs on vehicles generate energy, impacting consumption. The aim is to optimize operations including passenger service, charging, and vehicle-to-grid (V2G) interactions. Authors model fleet management using graphs and a linear program. Applied to a Gold Coast case, results show 10–15% lower costs with solar-electric fleets vs. electric-only. Larger V2G batteries reduce expenses through energy trading and potential profits. This research highlights solar's benefits in AMoD for cost savings and efficiency. The [212] paper optimizes the routing and charging infrastructure for Autonomous Mobility on Demand (AMoD) systems. With the advent of autonomy and electrical engineering, an AMoD system becomes possible. The authors propose a model for fleet and charging optimization. Using a mesoscopic approach, they create a time-invariant model that reflects routes and charging, taking into account vehicle charging. The road network is based on a digraph with isoenergetic arcs to achieve energy accuracy. The problem is a globally optimal mixed-integer linear program that is solved in less than 10 min, which is practical in the real world. Their approach is validated with case studies from taxis in New York City. Collaborative infrastructure optimization outperforms heuristics. Increasing the number of stations is not always beneficial.
- **Software optimization:** Software optimization is another important way to mitigate the environmental impact of self-driving vehicles. Authors of [213] confirmed that the energy consumption of edge processing reduces a car's mileage with up to 30%, making autonomous driving a difficult challenge for electric cars. Ref. [214] addresses the significant challenge of managing computational power consumption within autonomous vehicles, particularly for tasks involving artificial intelligence algorithms for sensing and perception. To overcome this challenge, the paper introduces an adaptive

optimization method that dynamically distributes onboard computational resources across multiple vehicular subsystems. This allocation is determined by the specific context in which the vehicle operates, accounting for factors like autonomous driving scenarios. By tailoring resource allocation to the situation, the proposed approach aims to enhance overall performance and reduce energy usage compared to conventional computational setups. The authors conducted experiments to validate their approach, creating diverse autonomous driving scenarios. The outcomes highlighted that their adaptive optimization method effectively enhanced both performance and energy efficiency in autonomous vehicles. This research makes a significant contribution by addressing a key hurdle in autonomous vehicle development—optimizing computational resource allocation using real-time contextual information. Overall, the paper presents an innovative strategy to tackle the issue of high computational power consumption in autonomous vehicles. Through dynamic resource allocation based on context, it offers potential enhancements in performance and energy efficiency. The findings yield valuable insights for optimizing computational resource allocation in autonomous vehicles and provide guidance for future research in this area. Moreover, reliance on multiple sensors, resource-intensive deep-learning models, and powerful hardware for safe navigation comes with challenges. Certain sensing modalities can hinder perception and escalate energy usage. To counteract this, the authors of [215] introduce EcoFusion—an energy-conscious sensor fusion approach that adapts fusion based on context, curtailing energy consumption while preserving perception quality. The proposed approach surpasses existing methods, improving object detection by up to 9.5%, while reducing energy consumption by around 60% and achieving a 58% latency reduction on Nvidia Drive PX2 hardware. Beyond EcoFusion, the authors suggest context identification strategies and execute a joint optimization for energy efficiency and performance. Context-specific results validate the efficacy of their approach.

7. Conclusions

In this survey, we have discussed the Features extraction, Mapping, and Localization processes in order to ensure safe localization and navigation in autonomous driving. Features represent the landmarks of the map, or, more precisely, what we want from the vehicle's environment (e.g., poles, curbs, buildings, intensity). The more relevant the features are the more accurate the mapping, and therefore the more accurate vehicle localization will be. This survey presents a novel classification including the semantic, non-semantic, and deep learning approaches to facilitate the comprehension of these methods. Our investigation concluded that using non-semantic features is more suitable to solve localization and mapping tasks. We also identified that the extraction of semantic features from the vehicles environment is time intensive and that such features do not exist in all environments. Deep learning methods are excellent tools to represent the environment semantically or not depending on the approach used. Although the performance of deep learning methods are high, the high computational cost makes it a bit difficult to execute.

Different mapping techniques have been investigated, including the pre-built and online ones and their underlying mathematical intuitions. We note that maps need to be frequently updated to reflect changes in the vehicles environment. Moreover, choosing the right features to include in the map is crucial to minimising computational memory use.

The last concept that should be demystified is localization, where we have tried to present, as much as possible, the methods that aim to localize the vehicles according to different sensors. The approaches based on probability, optimization, or cooperative localization are more robust and provide accurate results compared to the classical ones. Each approach was described briefly and recent related papers in the literature were investigated. We have also reviewed some of the attack methods that can threaten the AV systems in general.

In conclusion, our analysis offers an extensive investigation into the potential ramifications of autonomous driving. In addition, this study embraces a broad spectrum of considerations, encompassing areas such as energy usage reduction, emission management, noise and light pollution mitigation, seamless integration within smart urban landscapes, infrastructure optimization to facilitate self-driving technology, and the fine-tuning of software algorithms for optimal performance. By delving into these essential dimensions, our endeavor seeks to furnish a comprehensive understanding of the intricate and diverse repercussions associated with the widespread adoption of autonomous driving across various spheres of interest.

Author Contributions: Conceptualization, A.C., K.E.M. and U.O.; methodology, A.C.; software, A.Y.; validation, V.P. and E.U.E.; formal analysis, A.C.; resources, A.C.; data curation, A.C.; writing—original draft preparation, A.C. and E.U.E.; writing—review and editing, A.C., U.O. and E.U.E.; visualization, A.Y.; supervision, V.P., K.E.M. and A.Y.; project administration, K.E.M., A.Y. and V.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: We extend our heartfelt gratitude to all those who have contributed to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. *Road Traffic Injuries*; World Health Organization: Geneva, Switzerland, 2021. Available online: <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries> (accessed on 2 July 2021).
2. Du, H.; Zhu, G.; Zheng, J. Why travelers trust and accept self-driving cars: An empirical study. *Travel Behav. Soc.* **2021**, *22*, 1–9. [CrossRef]
3. Kopestinsky, A. 25 Astonishing Self-Driving Car Statistics for 2021. PolicyAdvice. 2021. Available online: <https://policyadvice.net/insurance/insights/self-driving-car-statistics/> (accessed on 2 July 2021).
4. Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *IEEE Access* **2020**, *8*, 58443–58469. [CrossRef]
5. Jo, K.; Kim, J.; Kim, D.; Jang, C.; Sunwoo, M. Development of autonomous car-part i: Distributed system architecture and development process. *IEEE Trans. Ind. Electron.* **2014**, *61*, 7131–7140. [CrossRef]
6. Blanco-Claraco, J.L. A Tutorial on SE(3) Transformation Parameterizations and On-Manifold Optimization. No. 3. 2021. Available online: <https://arxiv.org/abs/2103.15980> (accessed on 2 July 2021).
7. Sjafrie, H. *Introduction to Self-Driving Vehicle Technology*; Taylor Francis: Oxfordshire, UK, 2020.
8. Piasco, N.; Sidibé, D.; Demonceaux, C.; Gouet-Brunet, V. A survey on Visual-Based Localization: On the benefit of heterogeneous data. *Pattern Recognit.* **2018**, *74*, 90–109. [CrossRef]
9. Garcia-Fidalgo, E.; Ortiz, A. Vision-based topological mapping and localization methods: A survey. *Rob. Auton. Syst.* **2015**, *64*, 1–20. [CrossRef]
10. Grigorescu, S.; Trasnea, B.; Cocias, T.; Macesanu, G. A survey of deep learning techniques for autonomous driving. *J. Field Robot.* **2020**, *37*, 362–386. [CrossRef]
11. Wu, Y.; Wang, Y.; Zhang, S.; Ogai, H. Deep 3D Object Detection Networks Using LiDAR Data: A Review. *IEEE Sens. J.* **2021**, *21*, 1152–1171. [CrossRef]
12. Liu, W.; Sun, J.; Li, W.; Hu, T.; Wang, P. Deep learning on point clouds and its application: A survey. *Sensors* **2019**, *19*, 4188. [CrossRef]
13. Huang, B.; Zhao, J.; Liu, J. A Survey of Simultaneous Localization and Mapping with an Envision in 6G Wireless Networks. 2019, pp. 1–17. Available online: <https://arxiv.org/abs/1909.05214> (accessed on 5 July 2021).
14. Bresson, G.; Alsayed, Z.; Yu, L.; Glaser, S. Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving. *IEEE Trans. Intell. Veh.* **2017**, *2*, 194–220. [CrossRef]
15. Xia, L.; Cui, J.; Shen, R.; Xu, X.; Gao, Y.; Li, X. A survey of image semantics-based visual simultaneous localization and mapping: Application-oriented solutions to autonomous navigation of mobile robots. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1–17. [CrossRef]
16. Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: Part I. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110. [CrossRef]
17. Günay, F.B.; Öztürk, E.; Çavdar, T.; Hanay, Y.S.; Khan, A.U.R. *Vehicular Ad Hoc Network (VANET) Localization Techniques: A Survey*; Springer Netherlands: Dordrecht, The Netherlands, 2021; Volume 28. [CrossRef]
18. Kuutti, S.; Fallah, S.; Katsaros, K.; Dianati, M.; McCullough, F.; Mouzakitis, A. A Survey of the State-of-the-Art Localization Techniques and Their Potentials for Autonomous Vehicle Applications. *IEEE Internet Things J.* **2018**, *5*, 829–846. [CrossRef]
19. Badue, C.; Guidolini, R.; Carneiro, R.V.; Azevedo, P.; Cardoso, V.B.; Forechi, A.; Jesus, L.; Berriel, R.; Paixão, T.M.; Mutz, F.; et al. Self-driving cars: A survey. *Expert Syst. Appl.* **2021**, *165*, 113816. [CrossRef]

20. Betz, J.; Zheng, H.; Liniger, A.; Rosolia, U.; Karle, P.; Behl, M.; Krovi, V.; Mangharam, R. Autonomous vehicles on the edge: A survey on autonomous vehicle racing. *IEEE Open J. Intell. Transp. Syst.* **2022**, *3*, 458–488. [CrossRef]
21. Kim, D.; Chung, T.; Yi, K. Lane map building and localization for automated driving using 2D laser rangefinder. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Republic of Korea, 28 June–1 July 2015; pp. 680–685. [CrossRef]
22. Im, J.; Im, S.; Jee, G.-I. Extended Line Map-Based Precise Vehicle Localization Using 3D LiDAR. *Sensors* **2018**, *18*, 3179. [CrossRef]
23. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [CrossRef]
24. Zhang, Y.; Wang, J.; Wang, X.; Li, C.; Wang, L. A real-time curb detection and tracking method for UGVs by using a 3D-LiDAR sensor. In Proceedings of the 2015 IEEE Conference on Control Applications (CCA), Sydney, Australia, 21–23 September 2015; pp. 1020–1025. [CrossRef]
25. Wang, L.; Zhang, Y.; Wang, J. Map-Based Localization Method for Autonomous Vehicles Using 3D-LiDAR. *IFAC-PapersOnLine* **2017**, *50*, 276–281. [CrossRef]
26. Sefati, M.; Daum, M.; Sondermann, B.; Kreisköther, K.D.; Kampker, A. Improving vehicle localization using semantic and pole-like landmarks. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 13–19. [CrossRef]
27. Kummerle, J.; Sons, M.; Poggenhans, F.; Kuhner, T.; Lauer, M.; Stiller, C. Accurate and efficient self-localization on roads using basic geometric primitives. In Proceedings of the 2019 IEEE International Conference on Robotics and Automation (IEEE ICRA 2019), Montreal, QC, Canada, 20–24 May 2019; pp. 5965–5971. [CrossRef]
28. Zhang, C.; Ang, M.H.; Rus, D. Robust LiDAR Localization for Autonomous Driving in Rain. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2018, Madrid, Spain, 1–5 October 2018; pp. 3409–3415. [CrossRef]
29. Weng, L.; Yang, M.; Guo, L.; Wang, B.; Wang, C. Pole-Based Real-Time Localization for Autonomous Driving in Congested Urban Scenarios. In Proceedings of the 2018 IEEE International Conference on Real-time Computing and Robotics (RCAR), Kandima, Maldives, 1–5 August 2018; pp. 96–101. [CrossRef]
30. Lu, F.; Chen, G.; Dong, J.; Yuan, X.; Gu, S.; Knoll, A. Pole-based Localization for Autonomous Vehicles in Urban Scenarios Using Local Grid Map-based Method. In Proceedings of the 5th International Conference on Advanced Robotics and Mechatronics, ICARM 2020, Shenzhen, China, 18–21 December 2020; pp. 640–645. [CrossRef]
31. Schaefer, A.; Büscher, D.; Vertens, J.; Luft, L.; Burgard, W. Long-term vehicle localization in urban environments based on pole landmarks extracted from 3-D LiDAR scans. *Rob. Auton. Syst.* **2021**, *136*, 103709. [CrossRef]
32. Gim, J.; Ahn, C.; Peng, H. Landmark Attribute Analysis for a High-Precision Landmark-based Local Positioning System. *IEEE Access* **2021**, *9*, 18061–18071. [CrossRef]
33. Pang, S.; Kent, D.; Morris, D.; Radha, H. FLAME: Feature-Likelihood Based Mapping and Localization for Autonomous Vehicles. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 5312–5319. [CrossRef]
34. Yuming, H.; Yi, G.; Chengzhong, X.; Hui, K. Why semantics matters: A deep study on semantic particle-filtering localization in a LiDAR semantic pole-map. *arXiv* **2023**, arXiv:2305.14038v1.
35. Dong, H.; Chen, X.; Stachniss, C. Online Range Image-based Pole Extractor for Long-term LiDAR Localization in Urban Environments. In Proceedings of the 2021 European Conference on Mobile Robots (ECMR), Bonn, Germany, 31 August–3 September 2021. [CrossRef]
36. Shipitko, O.; Kibalov, V.; Abramov, M. Linear Features Observation Model for Autonomous Vehicle Localization. In Proceedings of the 16th International Conference on Control, Automation, Robotics and Vision, ICARCV 2020, Shenzhen, China, 13–15 December 2020; pp. 1360–1365. [CrossRef]
37. Shipitko, O.; Grigoryev, A. Ground vehicle localization with particle filter based on simulated road marking image. In Proceedings of the 32nd European Conference on Modelling and Simulation, Wilhelmshaven, Germany, 22–26 May 2018; pp. 341–347. [CrossRef]
38. Shipitko, O.S.; Abramov, M.P.; Lukoyanov, A.S. Edge Detection Based Mobile Robot Indoor Localization. International Conference on Machine Vision. 2018. Available online: <https://www.semanticscholar.org/paper/Edge-detection-based-mobile-robot-indoor-Shipitko-Abramov/51fd6f49579568417dd2a56e4c0348cb1bb91e78> (accessed on 17 July 2021).
39. Wu, F.; Wei, H.; Wang, X. Correction of image radial distortion based on division model. *Opt. Eng.* **2017**, *56*, 013108. [CrossRef]
40. Weng, L.; Gouet-Brunet, V.; Soheilian, B. Semantic signatures for large-scale visual localization. *Multimed. Tools Appl.* **2021**, *80*, 22347–22372. [CrossRef]
41. Hekimoglu, A.; Schmidt, M.; Marcos-Ramiro, A. Monocular 3D Object Detection with LiDAR Guided Semi Supervised Active Learning. 2023. Available online: <http://arxiv.org/abs/2307.08415v1> (accessed on 25 January 2024).
42. Hungar, C.; Brakemeier, S.; Jürgens, S.; Köster, F. GRAIL: A Gradients-of-Intensities-based Local Descriptor for Map-based Localization Using LiDAR Sensors. In Proceedings of the IEEE Intelligent Transportation Systems Conference, ITSC 2019, Auckland, New Zealand, 27–30 October 2019; pp. 4398–4403. [CrossRef]
43. Hungar, C.; Fricke, J.; Stefan, J.; Frank, K. Detection of Feature Areas for Map-based Localization Using LiDAR Descriptors. In Proceedings of the 16th Workshop on Posit. Navigat. and Communicat, Bremen, Germany, 23–24 October 2019. [CrossRef]
44. Gu, B.; Liu, J.; Xiong, H.; Li, T.; Pan, Y. Epcp-icp: A 6d vehicle pose estimation method by fusing the roadside LiDAR point cloud and road feature. *Sensors* **2021**, *21*, 3489. [CrossRef] [PubMed]

45. Burt, A.; Disney, M.; Calders, K. Extracting individual trees from LiDAR point clouds using treeseg. *Methods Ecol. Evol.* **2019**, *10*, 438–445. [[CrossRef](#)]
46. Charroud, A.; Yahyaouy, A.; Moutaouakil, K.E.; Onyekpe, U. Localisation and mapping of self-driving vehicles based on fuzzy K-means clustering: A non-semantic approach. In Proceedings of the 2022 International Conference on Intelligent Systems and Computer Vision (ISCV), Fez, Morocco, 8–19 May 2022.
47. Charroud, A.; Moutaouakil, K.E.; Yahyaouy, A.; Onyekpe, U.; Palade, V.; Huda, M.N. Rapid localization and mapping method based on adaptive particle filters. *Sensors* **2022**, *22*, 9439. [[CrossRef](#)]
48. Charroud, A.; Moutaouakil, K.E.; Yahyaouy, A. Fast and accurate localization and mapping method for self-driving vehicles based on a modified clustering particle filter. *Multimed. Tools Appl.* **2023**, *82*, 18435–18457. [[CrossRef](#)]
49. Zou, Y.; Wang, X.; Zhang, T.; Liang, B.; Song, J.; Liu, H. BRoPH: An efficient and compact binary descriptor for 3D point clouds. *Pattern Recognit.* **2018**, *76*, 522–536. [[CrossRef](#)]
50. Kiforenko, L.; Drost, B.; Tombari, F.; Krüger, N.; Buch, A.G. A performance evaluation of point pair features. *Comput. Vis. Image Underst.* **2016**, *166*, 66–80. [[CrossRef](#)]
51. Logoglu, K.B.; Kalkan, S.; Temize, A.I. CoSPAIR: Colored Histograms of Spatial Concentric Surflet-Pairs for 3D object recognition. *Rob. Auton. Syst.* **2016**, *75*, 558–570. [[CrossRef](#)]
52. Buch, A.G.; Kraft, D. Local point pair feature histogram for accurate 3D matching. In Proceedings of the 29th British Machine Vision Conference, BMVC 2018, Newcastle, UK, 3–6 September 2018. Available online: <https://www.reconcell.eu/files/publications/Buch2018.pdf> (accessed on 20 July 2021).
53. Zhao, H.; Tang, M.; Ding, H. HoPPF: A novel local surface descriptor for 3D object recognition. *Pattern Recognit.* **2020**, *103*, 107272. [[CrossRef](#)]
54. Wu, L.; Zhong, K.; Li, Z.; Zhou, M.; Hu, H.; Wang, C.; Shi, Y. Pptfh: Robust local descriptor based on point-pair transformation features for 3d surface matching. *Sensors* **2021**, *21*, 3229. [[CrossRef](#)]
55. Yang, J.; Zhang, Q.; Xiao, Y.; Cao, Z. TOLDI: An effective and robust approach for 3D local shape description. *Pattern Recognit.* **2017**, *65*, 175–187. [[CrossRef](#)]
56. Prakhya, S.M.; Lin, J.; Chandrasekhar, V.; Lin, W.; Liu, B. 3DHoPD: A Fast Low-Dimensional 3-D Descriptor. *IEEE Robot. Autom. Lett.* **2017**, *2*, 1472–1479. [[CrossRef](#)]
57. Hu, Z.; Qianwen, T.; Zhang, F. Improved intelligent vehicle self-localization with integration of sparse visual map and high-speed pavement visual odometry. *Proc. Inst. Mech. Eng. Part J. Automob. Eng.* **2021**, *235*, 177–187. [[CrossRef](#)]
58. Li, Y.; Hu, Z.; Cai, Y.; Wu, H.; Li, Z.; Sotelo, M.A. Visual Map-Based Localization for Intelligent Vehicles from Multi-View Site Matching. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 1068–1079. [[CrossRef](#)]
59. Ge, G.; Zhang, Y.; Jiang, Q.; Wang, W. Visual features assisted robot localization in symmetrical environment using laser slam. *Sensors* **2021**, *21*, 1772. [[CrossRef](#)] [[PubMed](#)]
60. DBow3. Source Code. 2017. Available online: <https://github.com/rmsalinas/DBow3> (accessed on 26 July 2021).
61. Holliday, A.; Dudek, G. Scale-invariant localization using quasi-semantic object landmarks. *Auton. Robots* **2021**, *45*, 407–420. [[CrossRef](#)]
62. Wikipedia. Convolutional Neural Network. Available online: https://en.wikipedia.org/wiki/Convolutional_neural_network (accessed on 28 July 2021).
63. Li, B.; Zhang, T.; Xia, T. Vehicle detection from 3D LiDAR using fully convolutional network. *Robot. Sci. Syst.* **2016**, *12*. [[CrossRef](#)]
64. Minemura, K.; Liau, H.; Monrroy, A.; Kato, S. LMNet: Real-time multiclass object detection on CPU using 3D LiDAR. In Proceedings of the 2018 3rd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS 2018), Singapore, 21–23 July 2018; pp. 28–34. [[CrossRef](#)]
65. Zhou, Y.; Tuzel, O. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4490–4499. [[CrossRef](#)]
66. Beltrán, J.; Guindel, C.; Moreno, F.M.; Cruzado, D.; García, F.; Escalera, A.D.L. BirdNet: A 3D Object Detection Framework from LiDAR Information. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 3517–3523. [[CrossRef](#)]
67. Barrera, A.; Guindel, C.; Beltrán, J.; García, F. BirdNet+: End-to-End 3D Object Detection in LiDAR Bird’s Eye View. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Virtual, 20–23 September 2020. [[CrossRef](#)]
68. Simon, M.; Milz, S.; Amende, K.; Gross, H.-M. Complex-YOLO: Real-time 3D Object Detection on Point Clouds. *arXiv* **2018**, arXiv:1803.06199. Available online: <https://arxiv.org/abs/1803.06199> (accessed on 27 July 2021).
69. Ali, W.; Abdelkarim, S.; Zidan, M.; Zahran, M.; Sallab, A.E. YOLO3D: End-to-end real-time 3D oriented object bounding box detection from LiDAR point cloud. *Lect. Notes Comput. Sci.* **2019**, *11131 LNCS*, 716–728. [[CrossRef](#)]
70. Te, G.; Zheng, A.; Hu, W.; Guo, Z. RGCNN: Regularized graph Cnn for point cloud segmentation. In Proceedings of the MM ’18—26th ACM International Conference on Multimedia, Seoul, Republic of Korea, 22–26 October 2018; pp. 746–754. [[CrossRef](#)]
71. Hua, B.S.; Tran, M.K.; Yeung, S.K. Pointwise Convolutional Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 984–993. [[CrossRef](#)]

72. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. PointCNN: Convolution on X-transformed points. *Adv. Neural Inf. Process. Syst.* **2018**, *2018*, 820–830.
73. Lan, S.; Yu, R.; Yu, G.; Davis, L.S. Modeling local geometric structure of 3D point clouds using geo-cnn. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 998–1008. [CrossRef]
74. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph Cnn for learning on point clouds. *ACM Trans. Graph.* **2019**, *38*, 1–12. [CrossRef]
75. Xu, Y.; Fan, T.; Xu, M.; Zeng, L.; Qiao, Y. SpiderCNN: Deep learning on point sets with parameterized convolutional filters. *Lect. Notes Comput. Sci.* **2018**, *11212 LNCS*, 90–105. [CrossRef]
76. Wang, D.Z.; Posner, I. Voting for voting in online point cloud object detection. In *Robotics: Science and Systems*; Sapienza University of Rome: Rome, Italy, 2015; Volume 11. [CrossRef]
77. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep learning on point sets for 3D classification and segmentation. In Proceedings of the—30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 21–26 July 2017; pp. 77–85. [CrossRef]
78. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **2017**, *2017*, 5100–5109.
79. Hao, D.; Xieyuanli, C.; Simo, S.; Cyrill, S. Online Pole Segmentation on Range Images for Long-term LiDAR Localization in Urban Environments. *arXiv* **2022**, arXiv:2208.07364v1.
80. Zhang, R.; Guo, Z.; Zhang, W.; Li, K.; Miao, X.; Cui, B.; Qiao, Y.; Gao, P.; Li, H. PointCLIP: Point Cloud Understanding by CLIP. [Abstract]. 2021. Available online: <http://arxiv.org/abs/2112.02413v1> (accessed on 28 July 2021).
81. Zhu, X.; Zhang, R.; He, B.; Zeng, Z.; Zhang, S.; Gao, P. PointCLIP V2: Adapting CLIP for Powerful 3D Open-world Learning. *arXiv* **2022**, arXiv:2211.11682v1.
82. Nichol, A.; Jun, H.; Dhariwal, P.; Mishkin, P.; Chen, M. Point-E: A System for Generating 3D Point Clouds from Complex Prompts. *arXiv* **2022**, arXiv:2212.08751v1.
83. Liu, V.; Vermeulen, J.; Fitzmaurice, G.; Matejka, J. 3DALL-E: Integrating Text-to-Image AI in 3D Design Workflows. In Proceedings of the Woodstock '18: ACM Symposium on Neural Gaze Detection, Woodstock, NY, USA, 3–5 June 2018; ACM: New York, NY, USA, 2018; p. 20. [CrossRef]
84. Kasten, Y.; Rahamim, O.; Chechik, G. Point-Cloud Completion with Pretrained Text-to-image Diffusion Models. *arXiv* **2023**, arXiv:2306.10533v1.
85. Chen, R.; Liu, Y.; Kong, L.; Zhu, X.; Ma, Y.; Li, Y.; Hou, Y.; Qiao, Y.; Wang, W. CLIP2Scene: Towards Label-efficient 3D Scene Understanding by CLIP. *arXiv* **2023**, arXiv:2301.04926v2.
86. Wong, K.; Gu, Y.; Kamijo, S. Mapping for autonomous driving: Opportunities and challenges. *IEEE Intell. Transp. Syst. Mag.* **2021**, *13*, 91–106. [CrossRef]
87. Seif, H.G.; Hu, X. Autonomous driving in the iCity—HD maps as a key challenge of the automotive industry. *Engineering* **2016**, *2*, 159–162. [CrossRef]
88. Gwon, G.P.; Hur, W.S.; Kim, S.W.; Seo, S.W. Generation of a precise and efficient lane-level road map for intelligent vehicle systems. *IEEE Trans. Veh. Technol.* **2017**, *66*, 4517–4533. [CrossRef]
89. HD Map for Autonomous Vehicles Market, Markets-and-Markets. 2021. Available online: <https://www.marketsandmarkets.com/Market-Reports/hd-map-autonomous-vehicle-market-141078517.html> (accessed on 15 August 2021).
90. Hausler, S.; Milford, M. P1-021: Map Creation, Monitoring and Maintenance for Automated Driving-Literature Review. 2021. Available online: <https://imoveaustralia.com/wp-content/uploads/2021/01/P1%E2%80%90021-Map-creation-monitoring-and-maintenance-for-automated-driving.pdf> (accessed on 7 September 2021).
91. Herrtwich, R. The Evolution of the HERE HD Live Map at Daimler. 2018. Available online: <https://360.here.com/the-evolution-of-the-hd-live-map> (accessed on 7 September 2021).
92. Chellapilla, K. 'Rethinking Maps for Self-Driving'. Medium. 2018. Available online: <https://medium.com/wovenplanetlevel5/https-medium-com-lyftlevel5-rethinking-maps-for-self-driving-a147c24758d6> (accessed on 6 September 2021).
93. Vardhan, H. HD Maps: New Age Maps Powering Autonomous Vehicles. Geo Spatial Word. 2017. Available online: <https://www.geospatialworld.net/article/hd-maps-autonomous-vehicles/> (accessed on 7 September 2021).
94. Dahlström, T. How Accurate Are HD Maps for Autonomous Driving and ADAS Simulation? 2020. Available online: <https://atlatec.de/en/blog/how-accurate-are-hd-maps-for-autonomous-driving-and-adas-simulation/> (accessed on 7 September 2021).
95. Schreier, M. Environment representations for automated on-road vehicles. *At-Automatisierungstechnik* **2018**, *66*, 107–118. [CrossRef]
96. Thrun, S. Probabilistic robotics. *Commun. ACM* **2002**, *45*, 52–57. [CrossRef]
97. Elfring, J.; Torta, E.; Molengraft, R.v. Particle Filters: A Hands-On Tutorial. *Sensors* **2021**, *21*, 438. [CrossRef] [PubMed]
98. Schumann, S. Why We're Mapping Down to 20 cm Accuracy on Roads. HERE 360 blog. 2014. Available online: <https://360.here.com/2014/02/12/why-were-mapping-down-to-20cm-accuracy-on-roads/> (accessed on 3 September 2021).
99. Waymo. Building Maps for a Self-Driving Car. 2016. Available online: <https://blog.waymo.com/2019/09/building-maps-for-self-driving-car.html> (accessed on 11 September 2021).
100. Tom-Tom, The Netherlands. HD Map—Highly Accurate Border-to-Border Model of the Road. 2017. Available online: <http://download.tomtom.com/open/banners/HD-Map-Product-Info-Sheet-improved-1.pdf> (accessed on 4 September 2021).

101. Liu, R.; Wang, J.; Zhang, B. High Definition Map for Automated Driving: Overview and Analysis. *J. Navig.* **2020**, *73*, 324–341. [CrossRef]
102. Sanborn. High Definition (HD) Maps for Autonomous Vehicles. 2019. Available online: <https://www.sanborn.com/highly-automated-driving-maps-for-autonomous-vehicles/> (accessed on 4 September 2021).
103. Ushr. Company Information. Available online: https://static1.squarespace.com/static/5a4d1c29017db266a358080\7/t/5ac25a1df950b77ed28e2d21/1522686495724/20180309_Snapsh\ot+Backgrounder_final.pdf (accessed on 4 September 2021).
104. Self-Driving Safety Report NVIDIA, US. 2021. Available online: <https://www.nvidia.com/en-us/self-driving-cars/safety-report/> (accessed on 4 September 2021).
105. ZENRIN. Maps to the Future. 2020. Available online: <http://www.zenrin-europe.com/> (accessed on 12 September 2021).
106. NavInfo. Overview. 2021. Available online: <https://navinfo.com/en/autonomousdriving> (accessed on 13 September 2021).
107. Korosec, K. This Startup Is Using Uber and Lyft Drivers to Bring Self-Driving Cars to Market Faster. *The Verge*, 2017. Available online: <https://www.theverge.com/2017/7/19/16000272/lvl5-self-driving-car-tesla-map-LiDAR> (accessed on 13 September 2021).
108. Atlatec. High Definition Maps for Autonomous Driving and Simulation. 2021. Available online: <https://atlatec.de/en/> (accessed on 13 September 2021).
109. Zhang, H.; Chen, N.; Fan, G.; Yang, D. An improved scan matching algorithm in SLAM. In Proceedings of the 6th International Conference on Systems and Informatics (ICSAI 2019), Shanghai, China, 2–4 November 2019; pp. 160–164. [CrossRef]
110. Jamil, F.; Iqbal, N.; Ahmad, S.; Kim, D.H. Toward accurate position estimation using learning to prediction algorithm in indoor navigation. *Sensors* **2020**, *20*, 4410. [CrossRef] [PubMed]
111. Joram, N. Dead Reckoning—A Nature-Inspired Path Integration That Made Us Discover the New World. *Medium*. 2021. Available online: <https://medium.com/geekculture/dead-reckoning-a-nature-inspired-path-integration-that-made-us-discover-the-new-world-ce67ee9d407d> (accessed on 26 September 2021).
112. Fuchs, C.; Aschenbruck, N.; Martini, P.; Wieneke, M. Indoor tracking for mission critical scenarios: A survey. *Pervasive Mob. Comput.* **2011**, *7*, 1–15. [CrossRef]
113. Wikipedia. True-Range Multilateration. 2021. Available online: https://en.wikipedia.org/wiki/True-range_multilateration (accessed on 23 September 2021).
114. Noureldin, A.; El-Shafie, A.; Bayoumi, M. GPS/INS integration utilizing dynamic neural networks for vehicular navigation. *Inf. Fusion* **2011**, *12*, 48–57. [CrossRef]
115. Malleswaran, M.; Vaidehi, V.; Saravanaselvan, A.; Mohankumar, M. Performance analysis of various artificial intelligent neural networks for GPS/INS Integration. *Appl. Artif. Intell.* **2013**, *27*, 367–407. [CrossRef]
116. Dai, H.f.; Bian, H.w.; Wang, R.y.; Ma, H. An INS/GNSS integrated navigation in GNSS denied environment using recurrent neural network. *Def. Technol.* **2020**, *16*, 334–340. [CrossRef]
117. Onyekpe, U.; Palade, V.; Kanarachos, S.; Christopoulos, S.R.G. Learning Uncertainties in Wheel Odometry for Vehicular Localisation in GNSS Deprived Environments. In Proceedings of the 19th IEEE International Conference on Machine Learning and Applications (ICMLA 2020), Miami, FL, USA, 14–17 December 2020; pp. 741–746. [CrossRef]
118. Onyekpe, U.; Palade, V.; Herath, A.; Kanarachos, S.; Fitzpatrick, M.E. WhONet: Wheel Odometry neural Network for vehicular localisation in GNSS-deprived environments. *Eng. Appl. Artif. Intell.* **2021**, *105*, 104421. [CrossRef]
119. Harvey, S.; Lee, A. Introducing: The Fingerprint Base Map™ for Autonomous Vehicle Mapping and Localization. *Medium*. 2018. Available online: <https://medium.com/@CivilMaps/introducing-the-fingerprint-base-map-for-autonomous-vehicle-mapping-and-localization-649dbd1e4810> (accessed on 26 September 2021).
120. Besl, P.J.; McKay, N.D. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [CrossRef]
121. Zhu, H.; Guo, B.; Zou, K.; Li, Y.; Yuen, K.V.; Mihaylova, L.; Leung, H. A review of point set registration: From pairwise registration to groupwise registration. *Sensors* **2019**, *19*, 1191. [CrossRef] [PubMed]
122. Wang, F.; Zhao, Z. A survey of iterative closest point algorithm. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; Volume 2017, pp. 4395–4399. [CrossRef]
123. Biber, P. The Normal Distributions Transform: A New Approach to Laser Scan Matching. *IEEE Int. Conf. Intell. Robot. Syst.* **2003**, *3*, 2743–2748. [CrossRef]
124. Silver, D. NDT Matching, *Medium*. 2017. Available online: <https://medium.com/self-driving-cars/ndt-matching-acff8e7e01cb> (accessed on 27 September 2021)
125. Shi, X.; Peng, J.; Li, J.; Yan, P.; Gong, H. The Iterative Closest Point Registration Algorithm Based on the Normal Distribution Transformation. *Procedia Comput. Sci.* **2019**, *147*, 181–190. [CrossRef]
126. Wikipedia. Point Set Registration. Available online: https://en.wikipedia.org/wiki/Point_set_registration (accessed on 27 September 2021).
127. Chui, H.; Rangarajan, A. A new point matching algorithm for non-rigid registration. *Comput. Vis. Image Underst.* **2003**, *89*, 114–141. [CrossRef]
128. Jian, B.; Vemuri, B.C. Robust Point Set Registration Using Gaussian Mixture Models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1633–1645. [CrossRef]

129. Yuan, W.; Eckart, B.; Kim, K.; Jampani, V.; Fox, D.; Kautz, J. DeepGMR: Learning Latent Gaussian Mixture Models for Registration. *Lect. Notes Comput. Sci.* **2020**, *12350 LNCS*, 733–750. [[CrossRef](#)]
130. Deng, H.; Birdal, T.; Ilic, S. 3D local features for direct pairwise registration. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 3239–3248. [[CrossRef](#)]
131. Aoki, Y.; Goforth, H.; Srivatsan, R.A.; Lucey, S. Pointnetlk: Robust & efficient point cloud registration using pointnet. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019), Long Beach, CA, USA, 15–20 June 2019; pp. 7156–7165. [[CrossRef](#)]
132. Wang, Y.; Solomon, J. Deep closest point: Learning representations for point cloud registration. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3522–3531. [[CrossRef](#)]
133. Huang, X.; Mei, G.; Zhang, J.; Abbas, R. A Comprehensive Survey on Point Cloud Registration. 2021, pp. 1–17. Available online: <https://arxiv.org/abs/2103.02690> (accessed on 10 August 2021).
134. Woo, A.; Fidan, B.; Melek, W.W. Localization for Autonomous Driving. In *Handbook of Position Location: Theory, Practice, and Advances*; John Wiley: Hoboken, NJ, USA, 2019; pp. 1051–1087. [[CrossRef](#)]
135. Jiang, Z.; Xu, Z.; Li, Y.; Min, H.; Zhou, J. Precise vehicle ego-localization using feature matching of pavement images. *J. Intell. Connect. Veh.* **2020**, *3*, 37–47. [[CrossRef](#)]
136. Sivic, J.; Zisserman, A. Video google: A text retrieval approach to object matching in videos. In Proceedings of the Ninth IEEE International Conference on Computer Vision 2003, Nice, France, 13–16 October 2003; Volume 2, pp. 1470–1477. [[CrossRef](#)]
137. Philbin, J.; Chum, O.; Isard, M.; Sivic, J.; Zisserman, A. Lost in quantization: Improving particular object retrieval in large scale image databases. In Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), Anchorage, AK, USA, 24–26 June 2008. [[CrossRef](#)]
138. Cao, S.; Snavely, N. Graph-based discriminative learning for location recognition. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 700–707. [[CrossRef](#)]
139. Aubry, M.; Russell, B.C.; Sivic, J. Alignment via Discriminative Visual Elements. 2014. Available online: <https://hal.inria.fr/hal-00863615v1/document> (accessed on 11 August 2021).
140. Lu, G.; Yan, Y.; Ren, L.; Song, J.; Sebe, N.; Kambhamettu, C. Localize Me Anywhere, Anytime: A Multi-task Point-Retrieval Approach. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 2434–2442. [[CrossRef](#)]
141. Arth, C.; Pirschheim, C.; Ventura, J.; Schmalstieg, D.; Lepetit, V. Instant Outdoor Localization and SLAM Initialization from 2.5D Maps. *IEEE Trans. Vis. Comput. Graph.* **2015**, *21*, 1309–1318. [[CrossRef](#)]
142. Poglitsch, C.; Arth, C.; Schmalstieg, D.; Ventura, J. [POSTER] A Particle Filter Approach to Outdoor Localization Using Image-Based Rendering. In Proceedings of the 2015 IEEE International Symposium on Mixed and Augmented Reality, Fukuoka, Japan, 29 September–3 October 2015; pp. 132–135. [[CrossRef](#)]
143. Song, Y.; Chen, X.; Wang, X.; Zhang, Y.; Li, J. 6-DOF Image Localization From Massive Geo-Tagged Reference Images. *IEEE Trans. Multimed.* **2016**, *18*, 1542–1554. [[CrossRef](#)]
144. Sattler, T.; Leibe, B.; Kobbelt, L. Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1744–1756. [[CrossRef](#)] [[PubMed](#)]
145. Shotton, J.; Glocker, B.; Zach, C.; Izadi, S.; Criminisi, A.; Fitzgibbon, A. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 2930–2937. [[CrossRef](#)]
146. Kendall, A.; Grimes, M.; Cipolla, R. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 2938–2946. [[CrossRef](#)]
147. Ahmed, S.Z.; Saputra, V.B.; Verma, S.; Zhang, K.; Adiwahono, A.H. Sparse-3D LiDAR Outdoor Map-Based Autonomous Vehicle Localization. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 1614–1619. [[CrossRef](#)]
148. Zhang, J.; Ai, D.; Xiang, Y.; Wang, Y.; Chen, X.; Chang, X. Bag-of-words based loop-closure detection in visual SLAM. In *Advanced Optical Imaging Technologies 2018*; SPIE: Bellingham, WA, USA, 2018; Volume 1081618, p. 45. [[CrossRef](#)]
149. Chen, Y.; Chen, Y.; Wang, G. Bundle Adjustment Revisited. 2019. Available online: <https://arxiv.org/abs/1912.03858> (accessed on 15 August 2021).
150. Li, J.; Pei, L.; Zou, D.; Xia, S.; Wu, Q.; Li, T.; Sun, Z.; Yu, W. Attention-SLAM: A Visual Monocular SLAM Learning from Human Gaze. *IEEE Sens. J.* **2021**, *21*, 6408–6420. [[CrossRef](#)]
151. Burri, M.; Nikolic, J.; Gohl, P.; Schneider, T.; Rehder, J.; Omari, S.; Achtelik, M.W.; Siegwart, R. The EuRoC micro aerial vehicle datasets. *Int. J. Rob. Res.* **2016**, *35*, 1157–1163. [[CrossRef](#)]
152. Mur-Artal, R.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [[CrossRef](#)]
153. Liu, Z.; Zhang, F. BALM: Bundle Adjustment for LiDAR Mapping. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3184–3191. [[CrossRef](#)]
154. Zhang, J.; Singh, S. LOAM: LiDAR Odometry and Mapping in Real-time. *Auton. Robots* **2014**, *41*, 401–416. [[CrossRef](#)]

155. Wang, K.; Ma, S.; Ren, F.; Lu, J. SBAS: Salient Bundle Adjustment for Visual SLAM. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–11. [CrossRef]
156. Lu, S.; Zhi, Y.; Zhang, S.; He, R.; Bao, Z. Semi-Direct Monocular SLAM with Three Levels of Parallel Optimizations. *IEEE Access* **2021**, *9*, 86801–86810. [CrossRef]
157. Grisetti, G.; Kummerle, R.; Stachniss, C.; Burgard, W. A tutorial on graph-based SLAM. *IEEE Intell. Transp. Syst. Mag.* **2010**, *2*, 31–43. [CrossRef]
158. Sualeh, M.; Kim, G.W. Simultaneous Localization and Mapping in the Epoch of Semantics: A Survey. *Int. J. Control. Autom. Syst.* **2019**, *17*, 729–742. [CrossRef]
159. Stachniss, C. Graph-Based SLAM in 90 Minutes. BONN University, 2020. Available online: https://www.unravel.rwth-aachen.de/global/show_document.asp?id=aaaaaaaaabebgwr&do\wnload=1 (accessed on 5 October 2021).
160. Mukherjee, S.; Kaess, M.; Martel, J.N.; Riviere, C.N. EyeSAM: Graph-based Localization and Mapping of Retinal Vasculature during Intraocular Microsurgery. *Physiol. Behav.* **2019**, *176*, 139–148. [CrossRef] [PubMed]
161. Jung, S.; Choi, D.; Song, S.; Myung, H. Bridge inspection using unmanned aerial vehicle based on HG-SLAM: Hierarchical graph-based SLAM. *Remote Sens.* **2020**, *12*, 3022. [CrossRef]
162. Jo, J.H.; Moon, C.B. Development of a Practical ICP Outlier Rejection Scheme for Graph-based SLAM Using a Laser Range Finder. *Int. J. Precis. Eng. Manuf.* **2019**, *20*, 1735–1745. [CrossRef]
163. Akca, A.; Efe, M.Ö. Multiple Model Kalman and Particle Filters and Applications: A Survey. *IFAC-PapersOnLine* **2019**, *52*, 73–78. [CrossRef]
164. Wadud, R.A.; Sun, W. DyOb-SLAM: Dynamic Object Tracking SLAM System. *arXiv* **2022**, arXiv:2211.01941.
165. Teed, Z.; Deng, J. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 16558–16569.
166. Li, Q.; Li, R.; Ji, K.; Dai, W. Kalman filter and its application. In Proceedings of the 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS 2015), Tianjin, China, 1–3 November 2015; pp. 74–77. [CrossRef]
167. Chen, S.Y. Kalman filter for robot vision: A survey. *IEEE Trans. Ind. Electron.* **2012**, *59*, 4409–4420. [CrossRef]
168. Li, W.; Wang, Z.; Yuan, Y.; Guo, L. Particle filtering with applications in networked systems: A survey. *Complex Intell. Syst.* **2016**, *2*, 293–315. [CrossRef]
169. Gustafsson, F. Particle filter theory and practice with positioning applications. *IEEE Aerosp. Electron. Syst. Mag.* **2010**, *25*, 53–82. [CrossRef]
170. Brossard, M.; Bonnabel, S.; Barrau, A. Invariant Kalman Filtering for Visual Inertial SLAM. In Proceedings of the 2018 21st International Conference on Information Fusion (FUSION), Cambridge, UK, 10–13 July 2018; pp. 2021–2028. [CrossRef]
171. Bahraini, M.S. On the Efficiency of SLAM Using Adaptive Unscented Kalman Filter. *Iran. J. Sci. Technol.-Trans. Mech. Eng.* **2020**, *44*, 727–735. [CrossRef]
172. Slowak, P.; Kaniewski, P. Stratified particle filter monocular SLAM. *Remote Sens.* **2021**, *13*, 3233. [CrossRef]
173. Zhang, Q.; Li, Y.; Ma, T.; Cong, Z.; Zhang, W. Bathymetric Particle Filter SLAM with Graph-Based Trajectory Update Method. *IEEE Access* **2021**, *9*, 85464–85475. [CrossRef]
174. Aghili, F. LiDAR SLAM for Autonomous Driving Vehicles. *Int. J. Robot. Res.* **2010**, *29*, 321–341. [CrossRef]
175. Hyowon, K.; Karl, G.; Lennart, S.; Sunwoo, K.; Henk, W. PMBM-based SLAM Filters in 5G mmWave Vehicular Networks. *arXiv* **2022**, arXiv:2205.02502v1.
176. Chiu, C.-Y. SLAM Backends with Objects in Motion: A Unifying Framework and Tutorial. In Proceedings of the 2023 American Control Conference (ACC), San Diego, CA, USA, 31 May–2 June 2023; pp. 1635–1642. [CrossRef]
177. Zeadally, S.; Guerrero, J.; Contreras, J. A tutorial survey on vehicle-to-vehicle communications. *Telecommun. Syst.* **2020**, *73*, 469–489. [CrossRef]
178. Chehri, A.; Quadar, N.; Saadane, R. Survey on localization methods for autonomous vehicles in smart cities. In Proceedings of the SCA '19: 4th International Conference on Smart City Applications, Casablanca, Morocco, 2–4 October 2019. [CrossRef]
179. Aoki, S.; Rajkumar, R.R. A-DRIVE: Autonomous Deadlock Detection and Recovery at Road Intersections for Connected and Automated Vehicles. Available online: <http://arxiv.org/abs/2204.04910v1> (accessed on 11 April 2022).
180. Piperigkos, N.; Lalos, A.S.; Berberidis, K. Graph Laplacian Diffusion Localization of Connected and Automated Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 12176–12190. [CrossRef]
181. Typaldos, P.; Papageorgiou, M.; Papamichail, I. Optimization-based path-planning for connected and non-connected Automated Vehicles. *Transp. Res. Part Emerg. Technol.* **2022**, *134*, 103487. [CrossRef]
182. Katriniok, A.; Rosarius, B.; Mähönen, P. Fully Distributed Model Predictive Control of Connected Automated Vehicles in Intersections: Theory and Vehicle Experiments. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 18288–18300. [CrossRef]
183. Ploeg, C.v.; Smit, R.; Siagkris-Lekkos, A.; Benders, F.; Silvas, E. Anomaly Detection from Cyber Threats via Infrastructure to Automated Vehicle. In Proceedings of the 2021 European Control Conference (ECC), Delft, The Netherlands, 29 June–2 July 2021; pp. 1788–1794. [CrossRef]
184. Soatti, G.; Nicoli, M.; Garcia, N.; Denis, B.; Raulefs, R.; Wymeersch, H. Implicit cooperative positioning in vehicular networks. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 3964–3980. [CrossRef]
185. Ghaleb, F.A.; Zainal, A.; Rassam, M.A.; Abraham, A. Improved vehicle positioning algorithm using enhanced innovation-based adaptive Kalman filter. *Pervasive Mob. Comput.* **2017**, *40*, 139–155. [CrossRef]

186. Shao, Z.; Li, W.; Wu, Y.; Shen, L. Multi-layer and multi-dimensional information based cooperative vehicle localization in highway scenarios. In Proceedings of the 2010 IEEE 12th International Conference on Communication Technology, Nanjing, China, 11–14 November 2010; pp. 567–571. [CrossRef]
187. Hoang, G.M.; Denis, B.; Harri, J.; Slock, D.T.M. Robust data fusion for cooperative vehicular localization in tunnels. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium, Los Angeles, CA, USA, 11–14 June 2017; pp. 1372–1377. [CrossRef]
188. Chen, H.; Luo, R.; Feng, Y. Improving Autonomous Vehicle Mapping and Navigation in Work Zones Using Crowdsourcing Vehicle Trajectories. 2022. Available online: <http://arxiv.org/abs/2301.09194v1> (accessed on 15 August 2021).
189. Klupacs, J.; Gostar, A.K.; Bab-Hadiashar, A.; Palmer, J.; Hoseinnezhad, R. Distributed Complementary Fusion for Connected Vehicles. In Proceedings of the 2022 11th International Conference on Control, Automation and Information Sciences (ICCAIS), Hanoi, Vietnam, 21–24 November 2022; pp. 316–321. [CrossRef]
190. Pham, M.; Xiong, K. A survey on security attacks and defense techniques for connected and autonomous vehicles. *Comput. Secur.* **2021**, *109*, 1–24. [CrossRef]
191. Kukkala, V.; Thiruloga, S.V.; Pasricha, S. Roadmap for Cybersecurity in Autonomous Vehicles. *IEEE Consum. Electron. Mag.* **2022**, *11*, 13–23. [CrossRef]
192. Yang, T.; Lv, C. Secure Estimation and Attack Isolation for Connected and Automated Driving in the Presence of Malicious Vehicles. *IEEE Trans. Veh. Technol.* **2021**, *70*, 8519–8528. [CrossRef]
193. Ahn, N.; Lee, D.H. Vehicle Communication using Hash Chain-based Secure Cluster. *arXiv* **2019**, arXiv:1912.12392.
194. Ahn, N.Y.; Lee, D.H. Physical Layer Security of Autonomous Driving: Secure Vehicle-to-Vehicle Communication in A Security Cluster. *arXiv* **2019**, arXiv:1912.06527.
195. Halabi, J.; Artail, H. A Lightweight Synchronous Cryptographic Hash Chain Solution to Securing the Vehicle CAN bus. In Proceedings of the 2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET), Beirut, Lebanon, 14–16 November 2018; pp. 1–6. [CrossRef]
196. Cho, K.T.; Shin, K.G. Fingerprinting electronic control units for vehicle intrusion detection. In Proceedings of the 25th USENIX Security Symposium, Austin, TX, USA, 10–12 August 2016; pp. 911–927. Available online: <https://www.usenix.org/system/files/conference/usenixsecurity16/sec16papercho.pdf> (accessed on 11 August 2021).
197. Regulus Cyber LTD. Tesla Model 3 Spoofed off the Highway—Regulus Navigation System Hack Causes Car to Turn on Its Own. Regulu. 2019. Available online: <https://www.regulus.com/blog/tesla-model-3-spoofed-off-the-highway-regulus-navigation-system-hack-causes-car-to-turn-on-its-own#:~:text=Actual%20control%20of%20the%20car,while%20driving`%20with%20NOA%20engaged> (accessed on 20 October 2021).
198. Stottelaar, B.G. Practical Cyber-Attacks on Autonomous Vehicles. 2015. Available online: <http://essay.utwente.nl/66766/> (accessed on 11 August 2021).
199. Shoukry, Y.; Martin, P.; Yona, Y.; Diggavi, S.; Srivastava, M. PyCRA: Physical challenge-response authentication for active sensors under spoofing attacks. In Proceedings of the CCS'15: The 22nd ACM Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; pp. 1004–1015. [CrossRef]
200. Petit, J.; Stottelaar, B.; Feiri, M.; Kargl, F. Remote Attacks on Automated Vehicles Sensors: Experiments on Camera and LiDAR. Blackhat.com. 2015; pp. 1–13. Available online: <https://www.blackhat.com/docs/eu-15/materials/eu-15-Petit-Self-Driving-And-Connected-Cars-Fooling-Sensors-And-Tracking-Drivers-wp1.pdf> (accessed on 11 August 2021).
201. Chontanawat, J. Relationship between energy consumption, CO₂ emission and economic growth in ASEAN: Cointegration and Causality Model. *Energy Rep.* **2020**, *6*, 660–665. [CrossRef]
202. Shen, K.; Ke, X.; Yang, F.; Wang, W.; Zhang, C.; Yuan, C. Numerical Energy Analysis of In-wheel Motor Driven Autonomous Electric Vehicles. *IEEE Trans. Transp. Electrification* **2023**, *9*, 3662–3676. [CrossRef]
203. Li, H.; Li, N.; Kolmanovsky, I.; Girard, A. Energy-Efficient Autonomous Driving Using Cognitive Driver Behavioral Models and Reinforcement Learning. In *AI-Enabled Technologies for Autonomous and Connected Vehicles*; Murphey, Y.L., Kolmanovsky, I., Watta, P., Eds.; Lecture Notes in Intelligent Transportation and Infrastructure; Springer: Cham, Switzerland, 2023. [CrossRef]
204. Gokhale, V.; Barrera, G.M.; Prasad, R.V. FEEL: Fast, Energy-Efficient Localization for Autonomous Indoor Vehicles. In Proceedings of the ICC 2021—IEEE International Conference on Communications, Montreal, QC, Canada, 14–18 June 2021; pp. 1–6. [CrossRef]
205. Zhen, H.; Mosharafian, S.; Yang, J.J.; Velni, J.M. Eco-driving trajectory planning of a heterogeneous platoon in urban environments. *IFAC-PapersOnLine* **2022**, *55*, 161–166. [CrossRef]
206. Coelho, M.C.; Guarnaccia, C. Driving information in a transition to a connected and autonomous vehicle environment: Impacts on pollutants, noise and safety. *Transp. Res. Procedia* **2020**, *45*, 740–746. [CrossRef]
207. Silva, Ó.; Cordera, R.; González-González, E.; Nogués, S. Environmental impacts of autonomous vehicles: A review of the scientific literature. *Sci. Total Environ.* **2022**, *830*, 154615. [CrossRef]
208. Chen, X.; Deng, Y.; Ding, H.; Qu, G.; Zhang, H.; Li, P.; Fang, Y. Vehicle as a Service (VaaS): Leverage Vehicles to Build Service Networks and Capabilities for Smart Cities. *arXiv* **2023**, arXiv:2304.11397.
209. Lim, H.; Taeihagh, A. Autonomous Vehicles for Smart and Sustainable Cities: An In-Depth Exploration of Privacy and Cybersecurity Implications. *Energies* **2018**, *11*, 1062. [CrossRef]
210. Samak, T.; Samak, C.; Kandhasamy, S.; Krovi, V.; Xie, M. AutoDRIVE: A Comprehensive, Flexible and Integrated Digital Twin Ecosystem for Autonomous Driving Research & Education. *Robotics* **2023**, *12*, 77. [CrossRef]

211. Paparella, F.; Hofman, T.; Salazar, M. Cost-optimal Fleet Management Strategies for Solar-electric Autonomous Mobility-on-Demand Systems. *arXiv* **2023**, arXiv:2305.18816.
212. Luke, J.; Salazar, M.; Rajagopal, R.; Pavone, M. Joint Optimization of Autonomous Electric Vehicle Fleet Operations and Charging Station Siting. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; pp. 3340–3347. [[CrossRef](#)]
213. Teraki. Autonomous Cars' Big Problem: The Energy Consumption of Edge Processing Reduces a Car's Mileage with up to 30%. 15 May 2019. Available online: <https://medium.com/@teraki/energy-consumption-required-by-edge-computing-reduces-a-autonomous-cars-mileage-with-up-to-30-46b6764ea1b7> (accessed on 10 October 2023).
214. Jambotkar, S.; Guo, L.; Jia, Y. Adaptive Optimization of Autonomous Vehicle Computational Resources for Performance and Energy Improvement. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 7594–7600. [[CrossRef](#)]
215. Malawade, A.V.; Mortlock, T.; Faruque, M.A. EcoFusion: Energy-aware adaptive sensor fusion for efficient autonomous vehicle perception. In Proceedings of the 59th ACM/IEEE Design Automation Conference (DAC '22), San Francisco, CA, USA, 10–14 July 2022; pp. 481–486. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.